AD NUMBER

AD851812

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

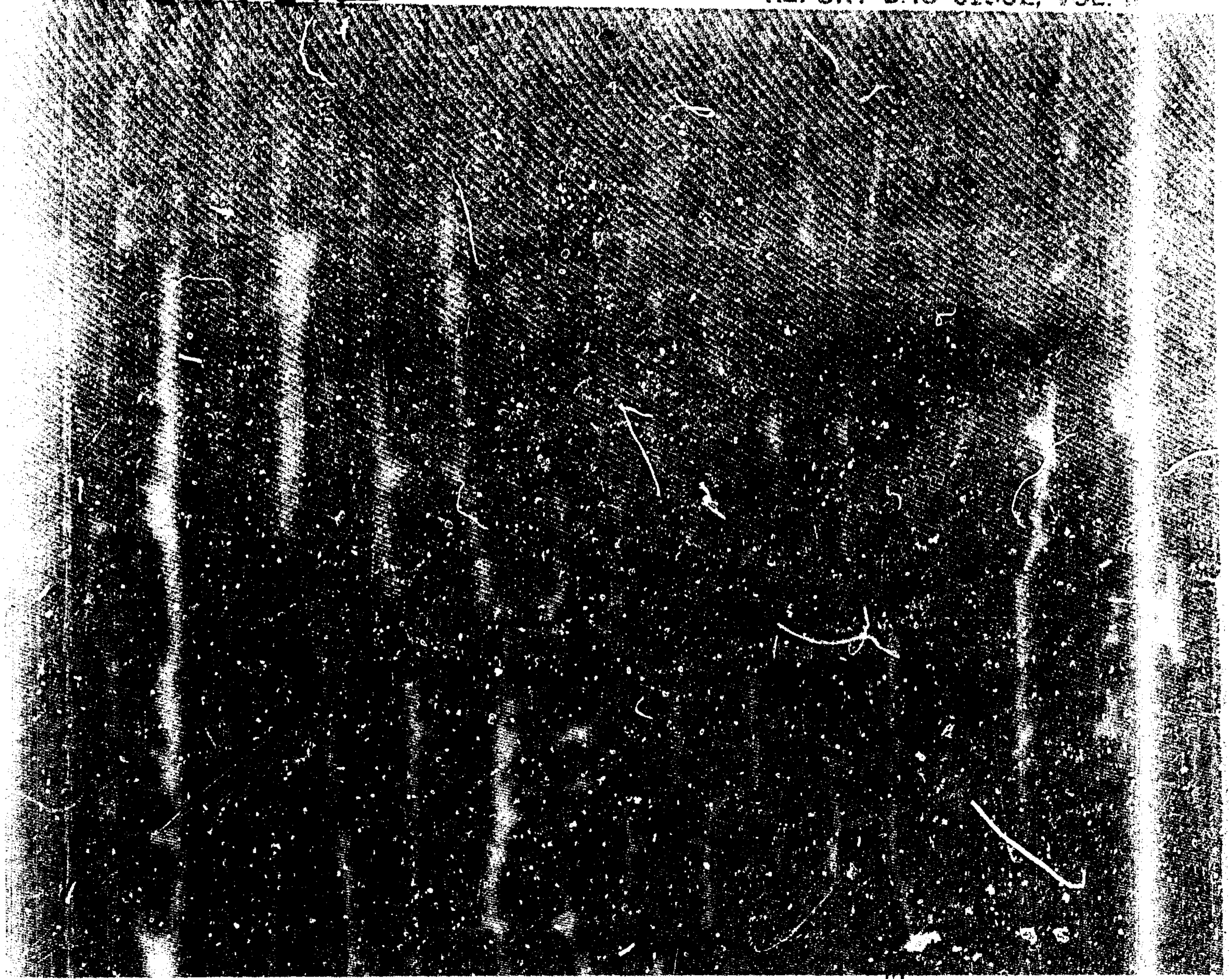
Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; APR 1968. Other requests shall be referred to Air Force Flight Dynamics Lab., Wright-Patterson AFB, OH 45433.

AUTHORITY

AFFDL ltr 9 Nov 1972

AD 851812

REPORT DAC 61552, VOL. II



HYPERSONIC ARBITRARY-BODY AERODYNAMIC COMPUTER PROGRAM (MARK III VERSION)

VOLUME II PROGRAM FORMULATION AND LISTINGS

BY

ARVEL E. GENTRY AND DOUGLAS N. SMYTH

APRIL 1968

MCDONNELL DOUGLAS CORPORATION

This Document Contains
Missing Page/s That Are
Unavailable In The
Original Document

OR are
Blank pgs.
that have
Been Removed

**BEST
AVAILABLE COPY**

HYPERSONIC ARBITRARY-BODY

AERODYNAMIC COMPUTER PROGRAM

MARK III VERSION

VOLUME II - PROGRAM FORMULATION AND LISTINGS

By

Arvel E. Gentry

and

Douglas N. Smyth

Douglas Report DAC 61552

April 1968

This material was prepared under sponsorship of the Douglas Independent Research and Development Program and Air Force Contracts No. F33615 67 C 1008 and F33615 67 C 1602. This report is provided in the interest of information exchange. Responsibility for the contents rests with the author or organization that prepared it.

DOUGLAS AIRCRAFT GROUP

3855 LAKEWOOD BOULEVARD, LONG BEACH, CALIFORNIA 90801

FOREWORD

This report describes a computer program developed at the Douglas Aircraft Division of the McDonnell-Douglas Corporation, Long Beach, California. The development of the Douglas Hypersonic Arbitrary-Body Aerodynamic Computer Program was started in 1964 and greatly expanded in subsequent years under sponsorship of the Douglas Independent Research and Development Program (IRAD). From August 1966 to May 1967 the program development was continued under Air Force Contract No. F3361567 C 1008. This contract was administered under the direction of the Aeronautical Systems Division, Directorate of Analysis, Wright-Patterson Air Force Base, Ohio by Mr. R. K. Mills, Project Engineer (ASBED-30). The product of this work was the Mark II version of the program as released for use by government agencies in May 1967. The latest version of the program as presented in this report (the Mark III version) is an extensively revised version of the earlier Mark II program. This version has been prepared as a result of both 1967-68 Douglas IRAD work and another Air Force contract (F33615 67 C 1602). This contract was administered by the Air Force Flight Dynamics Laboratory, Flight Mechanics Division, Gas Dynamics Branch, Mr. Valentine Dahlem, Project Engineer (FDMG).

At the Douglas Aircraft Division this work was conducted under the direction of Mr. A. E. Gentry as Principal Investigator. A number of people contributed to the various phases of this work for which the author is grateful. Mr. D. N. Smyth provided valuable consulting services in many phases of this work and prepared the new skin friction techniques incorporated in the Mark III version. Mr. W. R. Oliver's work in applying the various versions of this program to practical design problems contributed both in program design and in program validation. Others participating in this work include Messrs. G. D. Buell, J. L. Lundry, N. F. Wasson, and B. G. Wilson.

Special appreciation is extended to the various users of the earlier versions of this program for their valuable suggestions in a number of areas and for their efforts in adapting and running earlier versions of the program on the different types of computers. These include Messrs. Fred White, Jr. (Air Force ASBED-30), Don Shereda (Air Force FDMG), Ralph Carmichael and Charles Castolano (NASA Ames), C. L. W. Edwards (NASA Langley), Ralph Grahm (NASA Houston), Ray E. Aley (Lockheed Electronics Co., Houston), and R. E. Finch, A. W. Marziane, and J. H. Kainer (Aerospace Corp.).

This computer program and documentation report were released for general use by the author and by the Guidance and Control Section, ASBED-30, Wright-Patterson Air Force Base, in April 1968. This program and report are provided in the interest of information exchange. Responsibility for the contents rests with the author or organization that prepared it.

The dis ibution of computer program decks for the Mark III version is handled *he author.

ABSTRACT

This report describes a digital computer program system that is capable of calculating the hypersonic aerodynamic characteristics of complex three-dimensional shapes. The outstanding features of this program are its flexibility in covering a very wide variety of problems and the multitude of program options available. The program is a combination of techniques and capabilities necessary in performing a complete aerodynamic analysis of hypersonic shapes. These include vehicle geometry generation and description, visual graphics necessary in handling geometry data and in preparing plots of the final aerodynamic data, aerodynamic calculations of surface pressures and skin friction forces, and the integration of these forces to give all aerodynamic coefficients and stability derivatives.

The geometric description techniques in this program provide the capability of handling completely arbitrary three-dimensional shapes. The procedure developed to check the accuracy of the geometric data uses a computer and automatic recorder to draw pictures of the vehicle viewed from any angle.

The pressure calculation methods provided within the program include modified Newtonian, blunt-body Newtonian-Prandtl-Meyer, tangent-wedge, tangent-cone, shock-expansion, Prandtl-Meyer expansion, blast wave, modified tangent-cone, boundary-layer induced pressures, free-molecular flow, and a number of empirical relationships. The pressure calculation method most suitable for each component of the vehicle is specified by the aerodynamicist. Viscous forces are also calculated and include viscous-inviscid interaction effects. Skin friction options include the Reference Temperature and the Reference Enthalpy methods (for both laminar and turbulent flow), the Spalding-Chi method (turbulent), and a special blunt body skin friction method. Control surface deflection pressures, including separation effects that may be caused by the deflected surface, are also calculated.

The program has been used to study a wide variety of hypersonic vehicle shapes including hypersonic cruise aircraft, air-breathing booster aircraft, blunt lifting reentry bodies, high L/D reentry vehicles, blunt reentry capsules, rocket boosters, reentry warheads, and satellite shapes.

The program is documented in two volumes. Volume I is primarily a User's Manual, and Volume II contains the Program Formulation and Listings.

CONTENTS

CONTENTS (Continued)

ILLUSTRATIONS

SECTION I

INTRODUCTION

The objectives of the research work that led to this program were to (1) develop methods for determining the aerodynamic force characteristics of hypersonic vehicles regardless of the vehicle shape or flight condition, (2) program those techniques that required digital computer capability for practical and efficient application, and (3) verify these techniques by comparing the analytical results with test data.

At the start of this research project a list of guiding objectives was established to insure successful completion of the work. Major features desired in the final analysis system would:

1. Provide the ability to analyze completely arbitrary three-dimensional shapes.

2. Provide a component build-up capability where each vehicle component may be of arbitrary shape.

3. Include a number of force analysis methods so that the system would have the widest possible application to various vehicle shapes and flight conditions.

4. Provide the capability to use the best force calculation method for each vehicle component.

5. Provide methods for analyzing simple shapes within a minimum time period.

6. Develop a total analysis system framework that is adaptable to continued improvement and expansion.

The initial phase of this work was started in late 1964 and continued in 1965 as part of a Douglas Independent Research and Development Study. During that time a general arbitrary body force analysis approach was derived for hypersonic vehicles, the important basic components of the computer system were written and checked out, and the system demonstrated by application to several vehicles of completely arbitrary shape. All of this was accomplished under a very modest work effort.

During 1966 this work effort was expanded slightly and new capability added to the program system. This included the incorporation of several new force calculation methods, and the expansion of the force program to calculate vehicle static and dynamic stability derivatives.

In August of 1966, the program development was continued under Air Force Contract F33615 67 C 1008. This work included further expansion of the force calculation methods, the addition of new geometry description features, the incorporation of control surface derivative calculations, the consolidation of all the system components to form one large program, and

the preparation of complete program documentation information. The final program resulting from this work was identified as the Mark II version of the Hypersonic Arbitrary-Body Aerodynamic Computer Program.

During 1967 - 1968 the program development was continued under Douglas IRAD and another Air Force Contract (F33615 67 C 1602). During this period a number of major program modifications were accomplished including the addition of a number of new pressure calculation options, extensive revision to the skin friction parts of the program, and conversion of the program to operate on different computers. The program resulting from this work is identified as the Mark III version.

Throughout this report it will be assumed that the reader is familiar with the contents of Volume I, the User's Manual. Discussions of earlier versions of this program are given in References 1 and 2.

Both Volumes I and II of this report are essentially revised editions of the earlier Mark II program report (Reference 2). The differences between the Mark III and Mark II reports reflect the modifications and new capabilities provided by the latest version.

This report contains descriptions of the analysis techniques used within the program. Throughout these discussions an attempt has been made to maintain mathematical notations consistent with the appropriate references involved. This will assist the reader in comparing the approaches with the original reference material at some slight loss in continuity within the present report. This policy has also been used in the selection of many of the program variable names.

The program source language itself has been written in a manner so that the general flow of the program logic is easy to follow, even by the reader unfamiliar with FORTRAN. This has been accomplished by a very liberal use of comment statements throughout the program. This approach, together with the availability of machine-produced program flow charts, and the general widespread knowledge of the basic FORTRAN language, makes it unnecessary to include detailed equation-by-equation descriptions of the program content. Instead, it is only necessary to give a general mathematical description of the approach used.

Appendix A to this report contains the source language listings of the program, machine-produced flow charts, and the definitions of the program symbols. Appendix B contains one complete alphabetical listing of all the program variables and their definitions. Appendix C contains a description of the program subscripted variable arrays.

SECTION II

PROGRAM FORMULATION

PROBLEM FORMULATION

The problem of estimating the aerodynamic characteristics of arbitrary three-dimensional shapes at hypersonic speeds has several salient features. The first problem is to construct an accurate description of the vehicle geometry, as arbitrary and complex as it may be. This difficult geometry problem, together with the extremely wide range of flight conditions, dictates that many different force prediction techniques be available for use. The various approaches used in calculating the aerodynamic forces on three-dimensional shapes differ in that different methods are used to attack these two basic problems — the geometry representation problem, and the force calculation technique problem.

The most ambitious approach presently being attempted uses the method of characteristics. Some of this work is discussed in References 3 and 4. Undoubtedly, this is the eventual ideal approach to the calculation of forces on three-dimensional shapes at high speeds. However, present mathematical techniques and digital computer size and speed capabilities prevent application to typical preliminary design problems. Present applications of the method of characteristics must be reserved for simple bodies of revolution at zero angle of attack or important detail design applications where large computer times of several hours might be acceptable.

Other detailed gas dynamics approaches have been used for very simple shapes such as blunt-nosed bodies of revolution (see References 5 and 6). While this work has considerable value as an aid to understanding the chemical and gas dynamics problems associated with simple blunt shapes, its application to the general three-dimensional shape is restrictive.

Other special techniques, some theoretical and some empirical in nature, have been worked out for simple shapes. Notable examples are the work for hypersonic flow conditions on delta wings by Creager (Reference 7), and McLaughlin (Reference 8).

Another interesting approach to understanding the nature of high-speed flow is to select special shapes that are amenable to exact theory. In Reference 9 a class of delta wings is considered that permits solution using the exact shock wave theory. This work is expanded further in Reference 10 where upper surfaces are included and complete configurations suggested that are derived from simple shock waves and expansions.

Reference 11 contains an approach that uses parts of conical shock waves. This work is also discussed in Reference 12. A typical general approach to these and subsequent similar studies may be described as follows. Several exact methods have been derived for

the solution of flow fields about right circular cones at zero angle of attack. The shock wave system formed by this cone is also of a conical shape and the streamlines and flow properties of the air passing through this shock structure may be readily calculated (Reference 13). The first step in deriving a configuration is to define a wing leading edge line that lies on the surface of the shock wave cone. If we then follow the streamlines formed by the original cone from this leading edge line downstream, we find that we have described a surface along which solutions are already available from the exact cone flow calculations. Such an approach has been used in designing and evaluating some high speed airbreathing engine inlet designs. Although this approach again adds to the "storehouse" of knowledge on high speed aerodynamics, it certainly does not give much flexibility either in the design or the analysis process.

A more practical approach to the vehicle analysis problem was presented by Hankey in Reference 14. In this work it was recognized that each general type of vehicle shape (i.e., leading edge, flat surface, cone) required a different method for accurate force prediction. In this approach, solutions were obtained by restricting the vehicle to a few set combinations of simple geometrical shapes and by deriving theoretical and empirical force calculation methods for each shape component. The approach gave good results for the class of vehicles considered (the Dyna-Soar vehcile shape) but was limited in its application because of the restricted set of shapes available. The major contribution, however, was to illustrate that the method of force calculation must be tailored to the vehicle component shape and to the specific flight condition involved.

A similar approach has been in widespread use in both government and industry areas. The general procedure is to break the vehicle into a number of small simple shapes and to estimate the aerodynamic forces on each component by available methods. The widest and most frequently used force method is Newtonian theory. Several reports have been published that give results of Newtonian calculations for simple geometric shapes. References 15 and 16 contain design charts that are useful in evaluating certain geometric components (hemispheres, cylindrical leading edges, cones, cone frustums and flat plates) by Newtonian theory. The difficulty of this approach is that only simple shapes may be analyzed and only elementary force analysis method results such as given by Newtonian theory are available. For complex shapes considerable time and effort is required to define the vehicle in terms of required simple component shapes.

Reference 17 uses a semigraphical method for solving the arbitrary body geometry problem and uses Newtonian theory for the force calculations. The work by Dahlem (of ASD) modifies this procedure to allow computation on a digital computer (Reference 18). The geometric input data consist of coordinates of points on the surface, two angles indicating the surface slope and an angle indicating the orientation of the surface point. The inherent difficulty of preparing these input data for complex configurations from conventional engineering drawings is obvious.

A different approach to the geometry problem is presented by Van Tassell (of AVCO) in Reference 19. In this approach the body is approximated by a series of triangles on the body surface as determined from input coordinate points. In this case the force calculation methods include Newtonian theory and free-molecular flow (completely diffuse reflection). The use of surface coordinate points to form triangular elements still presents a difficult problem in obtaining accurate and correct input data.

A more elaborate technique of surface description has been worked out at MIT by Coons (Reference 20). In this method the surface is divided into a number of large patches and then each patch is mapped into a two-dimensional surface and described by a mathematical surface fit technique. The boundary curves of a patch are described by third order polynomials requiring only the coordinates and surface slopes at the four corner points in the transformed coordinate system. In a practical application the corner point surface slopes must be calculated by the computer since the problem of measuring them directly from engineering drawings would restrict the usefulness of this method. This may be accomplished by using a number of surface points along the boundaries to more completely describe the shape. After the boundary curves are defined in the transformed system the coordinates of a point on the surface may be calculated by using special blending relationships that serve as weighting functions to properly account for the influence of each boundary. Reference 21 contains a brief description of a similar technique used by the Boeing Company.

In most of the references cited above simple Newtonian theory was used to calculate the aerodynamic forces. While this method may give acceptable results for some shapes at very high Mach numbers, other methods must be available before a truly arbitrary body analysis system is available. An equally important problem is the transcription of the geometric description of the vehicle from engineering drawings and the preparation of this information in a form acceptable to the digital computer. Regardless of what surface description method is used, checking of the voluminous and often complex input data poses a difficult problem. Since a single vehicle may be composed of several different classes of general shapes (such as a blunt nose body, together with a thin wing) the arbitrary body analysis system must have the capability of using different force analysis techniques and frequently a different grid work of input surface data on each component of the vehicle.

For hypersonic speeds the most frequently used force analysis method is Newtonian theory. In this theory the pressure on each surface point of the vehicle is only a function of the angle that the surface makes with the free stream flow. Several other methods for calculating inviscid pressure also depend only on the local inclination of the surface. These include tangent-wedge and tangent-cone, and several other empirical methods worked out for special uses such as Newtonian plus Prandtl-Meyer expansion. Another important method for use at high speeds is hypersonic shock-expansion.

With the foregoing discussion as background information we will now proceed with the approach used to solve the general problem that is the subject of this research project — the development of methods for calculating the hypersonic, aerodynamic characteristics of completely arbitrary three-dimensional shapes.

In the following pages we will first present a brief introduction to the program organization. This will be followed by a more detailed derivation of the analysis methods used in each component of the system. The computer program system described in the following pages is capable of calculating the aerodynamic forces on completely arbitrary three-dimensional shapes at high supersonic and hypersonic speeds.

All program components of this system are written in FORTRAN.

PROGRAM COMPONENTS

This program is written on a completely modular basis to facilitate checkout and modification activities. The Mark III program contains five major program components: the Aerodynamic Program, the Picture Drawing Program, the Output Data Plotter Program, the Auxiliary Geometry Generation Program (Slab Delta Program), and the card punch routine.

In the early development phase each of these components was actually a completely separate computer program. The programs could, however, be run back-to-back with the output from one program saved on tape for use by the next. In the Mark III program all of these components are combined under the control of a small executive main program. However, to simplify the discussions in this report, each of these components is still referred to as a program even though it is really just a subroutine in the overall program.

The general activities carried out by the Arbitrary-Body Program System are involved with one of three basic tasks: (1) the preparation of geometry data, (2) the calculation of aerodynamic characteristics, and (3) the preparation of graphic output data. The computations performed in support of each of these system tasks are discussed in the following sections of this report.

GEOMETRY

This program contains several different methods for describing three-dimensional geometric shapes. These methods provide the flexibility required to analyze a variety of shapes ranging from very simple surfaces to the most complex forms. The program geometry options provided are (1) the surface element method, (2) the elliptical surface generation method, (3) the parametric cubic method, and (4) the slab delta geometry generation method. If desired, all of these methods could be used in describing a single vehicle shape.

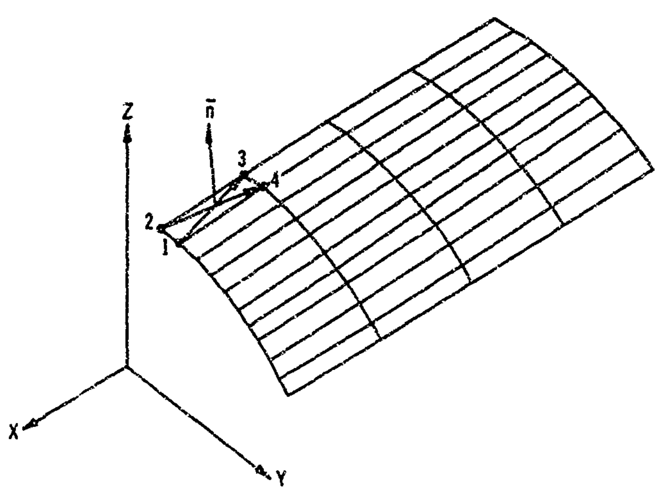
The principles involved in the application of each of these geometry methods are discussed in detail in the User's Manual and need not be covered here. The principal mathematical techniques, however, are

important from the programming standpoint and will be discussed on the following pages.

The Surface Element Geometry Method

The basic geometry method used by this program is the surface element or quadrilateral method. This method was developed by J. L. Hess and A. M. O. Smith for the Douglas Three-Dimensional Potential Flow Program (Reference 22). For completeness, certain parts of this report will be included in the following discussions.

The coordinate system used for this analysis is a right-handed Cartesian system as shown in the figure below.



DIAGONAL VECTORS $\bar{T}_1$ and $\bar{T}_2$

$T_{1x} = X_3 - X_1 \quad T_{1y} = Y_3 - Y_1 \quad T_{1z} = Z_3 - Z_1$

$T_{2x} = X_4 - X_2 \quad T_{2y} = Y_4 - Y_2 \quad T_{2z} = Z_4 - Z_2$

UNIT NORMAL $\quad \bar{N} = \bar{T}_2 \times \bar{T}_1$

$N_x = T_{2y}T_{1z} - T_{1y}T_{2z} \qquad n_x = N_x/N$

$N_y = T_{1x}T_{2z} - T_{2x}T_{1z} \qquad n_y = N_y/N$

$N_z = T_{2x}T_{1y} - T_{1x}T_{2y} \qquad n_z = N_z/N$

$$N = \sqrt{N_x^2 + N_y^2 + N_z^2}$$

AVERAGE POINT

$\bar{x} = \frac{1}{4}(X_1 + X_2 + X_3 + X_4)$

$\bar{y} = \frac{1}{4}(Y_1 + Y_2 + Y_3 + Y_4)$

$\bar{z} = \frac{1}{4}(Z_1 + Z_2 + Z_3 + Z_4)$

CORNER POINT PROJECTION DISTANCE

$d_k = n_x(\bar{x} - X_k) + n_y(\bar{y} - Y_k) + n_z(\bar{z} - Z_k)$

$k = 1, 2, 3, 4$

CORNER POINT COORDINATES

$X_k' = X_k + n_x d_k$

$Y_k' = Y_k + n_y d_k$

$Z_k' = Z_k + n_z d_k$

In the conventional use of this program the vehicle is usually positioned with its nose at the coordinate system origin and with the length of the body stretching in the negative X direction. The slight inconvenience of this negative sign on the body stations has been accepted so that the geometric data will be compatible with the Douglas potential flow program (Neumann Program).

The body surface is represented by a set of points in space. These points are selected on the body surface and are used by the method to obtain an approximation to this surface that is used in subsequent calculations. If the four related points of each set are connected by straight lines we may obtain a picture of how the input surface points are organized to describe a given shape. This has been done in Figure 1. The input scheme has been designed so that each point need only be input once even though it may be a member of as many as four adjacent sets of points. This is accomplished by the use of an additional parameter for each point besides the X, Y, and Z values. This parameter (known as the status flag) indicates whether a point is a continuation of a column of points (STATUS = 0), the beginning of a

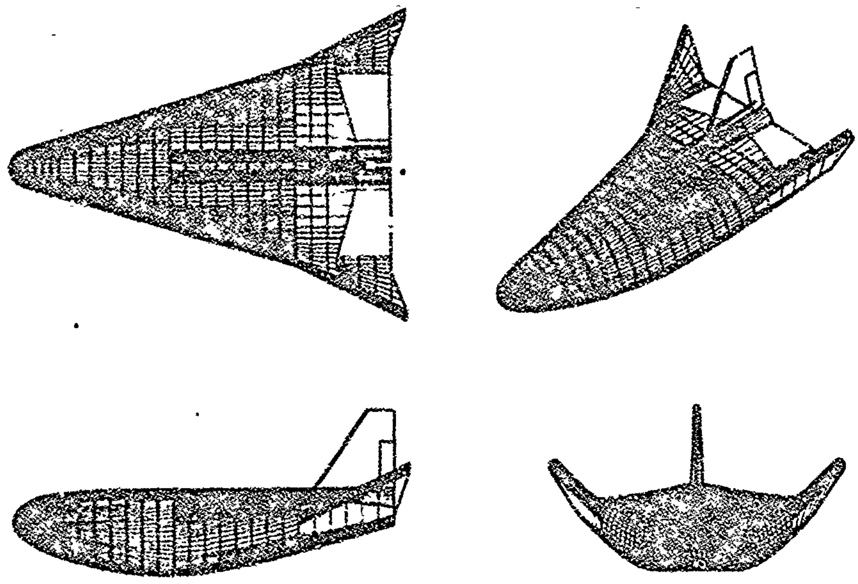


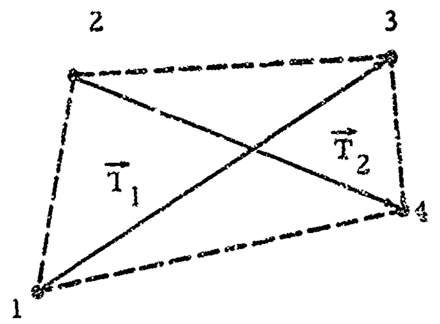
Figure 1. Output from Perspective Drawing Program

new column of points (=1), the first point of a new section of elements (=2), or the last point input for the shape (=3).

As may be seen from the drawings made by the Picture Drawing Program, the different areas of a vehicle may require a different organization and spacing of surface points for accurate representation. Each such area or organization of elements is called a section and each section is independent of all other sections. The division of a vehicle into a given set of sections may also be influenced by another consideration since the force calculation program may be made to calculate the force contributions of each section separately, using different calculation methods.

The input surface points are not sufficient in themselves for the force calculations. Each set of four related points which form an individual element must be converted into quantities useful to the program. This is accomplished by approximating each element area of the vehicle by a plane quadrilateral surface. Since we are using four surface points to form an element, no single surface will contain the points themselves. Also, adjacent plane quadrilateral edges will not necessarily be coincident. With a sufficiently small size of the surface elements this will be of no consequence in the end results.

The mathematical technique used in converting an input set of four points into a plane quadrilateral element is described below. The figure below gives a representation of the input element points with each point identified consecutively around the element by the subscripts 1, 2, 3, and 4, respectively.



The coordinates in the reference coordinate system are as follows:

1	:	x_1^i	y_1^i	z_1^i
2	:	x_2^i	y_2^i	z_2^i
3	:	x_3^i	y_3^i	z_3^i
4	:	x_4^i	y_4^i	z_4^i

The superscript i identifies the coordinates as input coordinates. We next form the two diagonal vectors $\vec{T_1}$ and $\vec{T_2}$. The components of these vectors are

$$T_{1x} = x_3^i - x_1^i \qquad T_{1y} = y_3^i - y_1^i \qquad T_{1z} = z_3^i - z_1^i$$

$$T_{2x} = x_4^i - x_2^i \qquad T_{2y} = y_4^i - y_2^i \qquad T_{2z} = z_4^i - z_2^i$$

We may now obtain a new vector $\vec{N}$ (and its components) by taking the cross product of the diagonal vectors.

$$\bar{N} = \vec{T}_2 \times \vec{T}_1$$

$$N_x = T_{2y}\, T_{1z} - T_{1y}\, T_{2z}$$

$$N_y = T_{1x}\, T_{2z} - T_{2x}\, T_{1z}$$

$$N_z = T_{2x}\, T_{1y} - T_{1x}\, T_{2y}$$

The unit normal vector, $\vec{n}$, to the plane of the element is taken as $\vec{N}$ divided by its own length N (direction cosines of outward unit normal).

$$n_x = \frac{N_x}{N}$$

$$n_y = \frac{N_y}{N}$$

$$n_z = \frac{N_z}{N}$$

where

$$N = \sqrt{N_x^{\,2} + N_y^{\,2} + N_z^{\,2}}$$

The plane of the element is now completely determined if a point in this plane is specified. This point is taken as the point whose coordinates, $\bar{x}$, $\bar{y}$, $\bar{z}$ are the averages of the coordinates of the four input points.

$$\bar{x} = \frac{1}{4}\left[x_1^i + x_2^i + x_3^i + x_4^i\right]$$

$$\bar{y} = \frac{1}{4}\left[y_1^i + y_2^i + y_3^i + y_4^i\right]$$

$$\bar{z} = \frac{1}{4}\left[z_1^i + z_2^i + z_3^i + z_4^i\right]$$

Now the input points will be projected into the plane of the element along the normal vector. The resulting points are the corner points of the quadrilateral element. The signed distance of the k-th input points (k = 1, 2, 3, 4) from the plane is

$$d_k = n_x(\bar{x} - x_k^i) + n_y(\bar{y} - y_k^i) + n_z(\bar{z} - z_k^i) \qquad k = 1, 2, 3, 4$$

It turns out that, due to the way in which the plane was generated from the input points, all the d_k's have the same magnitude, those for points 1 and 3 having one sign and those for points 2 and 4 having the opposite sign. Symbolically,

$$d_k = (-1)^{k-1} d_1 \qquad k = 1, 2, 3, 4$$

The magnitude of the common projection distance is called d, i.e.,

$$d = \left| d_1 \right|$$

The coordinates of the corner points in the reference coordinate system are given by

$$x_k' = x_k^i + n_x d_k$$

$$y_k' = y_k^i + n_y d_k \qquad k = 1, 2, 3, 4$$

$$z_k' = z_k^i + n_z d_k$$

Now the element coordinate system must be constructed. This requires the components of three mutually perpendicular unit vectors, one of which points along each of the coordinate axes of the system, and also the coordinates of the origin of the coordinate system. All these quantities must be given in terms of the reference coordinate system. The unit normal vector is taken as one of the unit vectors, so two perpendicular unit vectors in the plane of the element are needed. Denote these unit vectors $\vec{t}_1$ and $\vec{t}_2$. The vector $\vec{t}_1$ is taken as $\vec{T}_1$ divided by its own length T_1, i.e.,

$$t_{1x} = \frac{T_{1x}}{T_1}$$

$$t_{1y} = \frac{T_{1y}}{T_1}$$

$$t_{1z} = \frac{T_{1z}}{T_1}$$

where

$$T_1 = \sqrt{T_{1x}^2 + T_{1y}^2 + T_{1z}^2}$$

The vector $\vec{t_2}$ is defined by $\vec{t_2} = \vec{n} \times \vec{t_1}$, so that its components are

$$t_{2x} = n_y t_{1z} - n_z t_{1y}$$

$$t_{2y} = n_z t_{1x} - n_x t_{1z}$$

$$t_{2z} = n_x t_{1y} - n_y t_{1x}$$

The vector $\vec{t_1}$ is the unit vector parallel to the x or ξ axis of the element coordinate system, while $\vec{t_2}$ is parallel to the y or η axis, and n is parallel to the z or ζ axis of this coordinate system.

To transform the coordinates of points and the components of vectors between the reference coordinate system and the element coordinate system, the transformation matrix is required. The elements of this matrix are the components of the three basic unit vectors, $\vec{t_1}$, $\vec{t_2}$, and $\vec{n}$. To make the notation uniform define

$$a_{11} = t_{1x} \qquad a_{12} = t_{1y} \qquad a_{13} = t_{1z}$$

$$a_{21} = t_{2x} \qquad a_{22} = t_{2y} \qquad a_{23} = t_{2z}$$

$$a_{31} = n_x \qquad a_{32} = n_y \qquad a_{33} = n_z$$

The transformation matrix is thus the array

$$\begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{matrix}$$

To transform the coordinates of points from one system to the other, the coordinates of the origin of the element coordinate system in the reference coordinate system are required. Let these be denoted x_o, y_o, z_o. Then if a point has coordinates x', y', z' in the reference coordinate system and coordinates x, y, z in the element coordinate

system, the transformation from the reference to the element system is

$$x = a_{11}(x' - x_o) + a_{12}(y' - y_o) + a_{13}(z' - z_o)$$

$$y = a_{21}(x' - x_o) + a_{22}(y' - y_o) + a_{23}(z' - z_o)$$

$$z = a_{31}(x' - x_o) + a_{32}(y' - y_o) + a_{33}(z' - z_o)$$

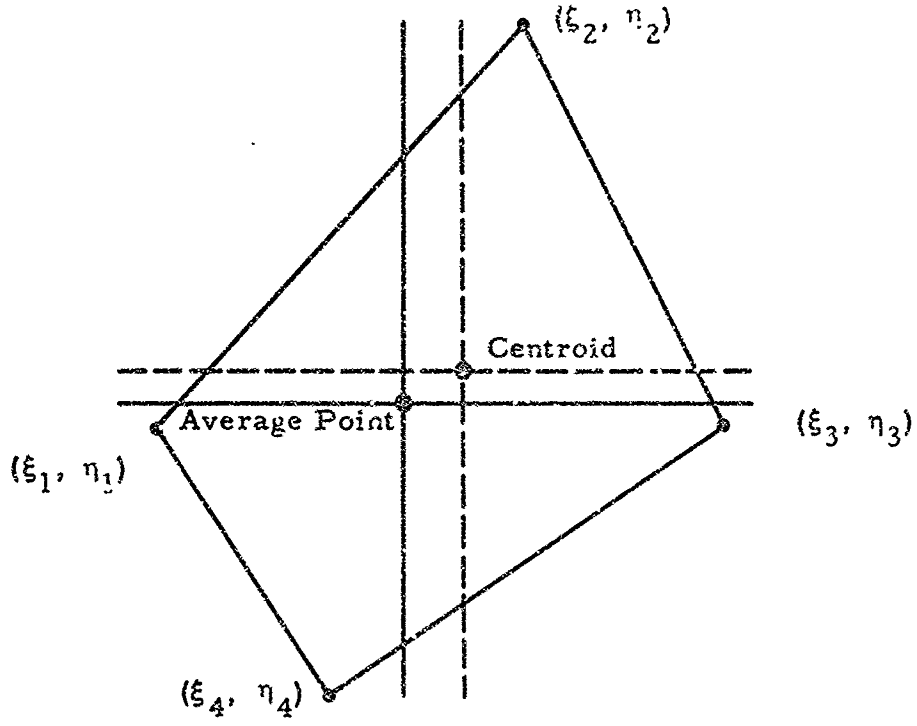
while the transformation from the element to the reference system is

$$x' = x_o + a_{11}x + a_{21}y + a_{31}z$$

$$y' = y_o + a_{12}x + a_{22}y + a_{32}z$$

$$z' = z_o + a_{13}x + a_{23}y + a_{33}z$$

The corner points are now transformed into the element coordinate system based on the average point as origin. These points have coordinates x'_k, y'_k, z'_k in the reference coordinate system. Their coordinates in the element coordinate system with this origin are denoted by ξ^*_k, η^*_k, 0. Because they lie in the plane of the element, they have a zero z or ζ coordinate in the element coordinate system. Also, because the vector $\vec{t}_1$, which defines the x or ξ axis of the element coordinate system, is a multiple of the "diagonal" vector from point 1 to point 3, the coordinate η^*_1 and the coordinate η^*_3 are equal. This is illustrated in the figure below. Using the above transformation these coordinates are explicitly



$$\xi_k^* = a_{11}(x_k' - \bar{x}) + a_{12}(y_k' - \bar{y}) + a_{13}(z_k' - \bar{z})$$

$$K = 1, 2, 3, 4$$

$$\eta_k^* = a_{21}(x_k' - \bar{x}) + a_{22}(x_k' - \bar{y}) + a_{23}(z_k' - \bar{z})$$

These corner points are taken as the corners of a plane quadrilateral.

The origin of the element coordinate system is now transferred to the centroid of the area of the quadrilateral. With the average point as origin the coordinates of the centroid in the element coordinate system are:

$$\xi_o = \frac{1}{3}\,\frac{1}{\eta_2^* - \eta_4^*}\left[\xi_4^*\,(\eta_1^* - \eta_2^*) + \xi_2^*\,(\eta_4^* - \eta_1^*)\right]$$

$$\eta_o = -\frac{1}{3}\eta_1^*$$

These are subtracted from the coordinates of the corner points in the element coordinate system based on the average point as origin to obtain the coordinates of the corner points in the element coordinate system based on the centroid as origin. Accordingly, these latter coordinates are

$$\xi_k = \xi_k^* - \xi_o \qquad\qquad K = 1, 2, 3, 4$$

$$\eta_k = \eta_k^* - \eta_o$$

Since the centroid is to be used as the origin of the element coordinate system, its coordinates in the reference coordinate system are required for use with the transformation matrix. These coordinates are

$$x_o = \bar{x} + a_{11}\,\xi_o + a_{21}\,\eta_o$$

$$y_o = \bar{y} + a_{12}\,\xi_o + a_{22}\,\eta_o$$

$$z_o = \bar{z} + a_{13}\,\xi_o + a_{23}\,\eta_o$$

Since in all subsequent transformations between the reference coordinate system and the element coordinate system the centroid is used as origin of the latter, its coordinates are denoted x_o, y_o, z_o. The coordinates of the average point are no longer needed. The change in origin of the element coordinate system, of course, has no effect on the coordinates of the corner points in the reference coordinate system.

The lengths of the two diagonals of the quadrilateral, t_1 and t_2, are computed from

$$t_1^2 = (\xi_3 - \xi_1)^2$$

$$t_2^2 = (\xi_4 - \xi_2)^2 + (\eta_4 - \eta_2)^2$$

The larger of these is selected and designated the maximum diagonal t.

The body surface area and enclosed volume are determined by summing up the contributions of each element. In terms of the coordinates of the corner points, the area of the quadrilateral is

$$A = \frac{1}{2}\,(\xi_3 - \xi_1)(\eta_2 - \eta_4)$$

The incremental volume is given by the volume of the parallepepiped formed by the element and its projection onto the x - z plane (the x - y or y - z planes would have served equally well).

$$V = y_o\, A\, n_y$$

Summary

The foregoing procedure may be briefly summarized as follows:

Each set of four points is converted into a plane-quadrilateral element by the procedure shown in the sketch on page 7. The normal to the quadrilateral is taken as the cross product of two diagonal vectors formed between opposite element points. The order of the input points and the manner of defining the diagonal vectors is used to ensure that the cross product gives an outward normal to the body surface. The next step is to define the plane of the element by determining the averages of the coordinates of the original four corner points. These points are then projected parallel to the normal vector into the plane of the element to give the corners of the plane quadrilateral. The corner points of the quadrilateral are equidistant from the four points used to form the element. Additional parameters required for subsequent force calculations, quadrilateral area and centroid, may now be calculated.

The spacing and orientation of the elements is varied in such a way that they describe the vehicle shape accurately. Since four points are used to define the plane quadrilateral, the edges of adjacent elements are not coincident. This is not important, since the pressure is calculated only at the quadrilateral centroid. This pressure is then assumed to be constant over the surface of the element.

The plane-quadrilateral surface description method is not as elaborate as some of the other methods. It is important, however, to note that the simplicity of the method permits the use of conventional cross-sectional drawings in data preparation (no surface slopes required) and the use of semiautomatic data-reading techniques. Also, as has been illustrated in Volume I, computer-generated pictures are used in checking the geometric data for errors.

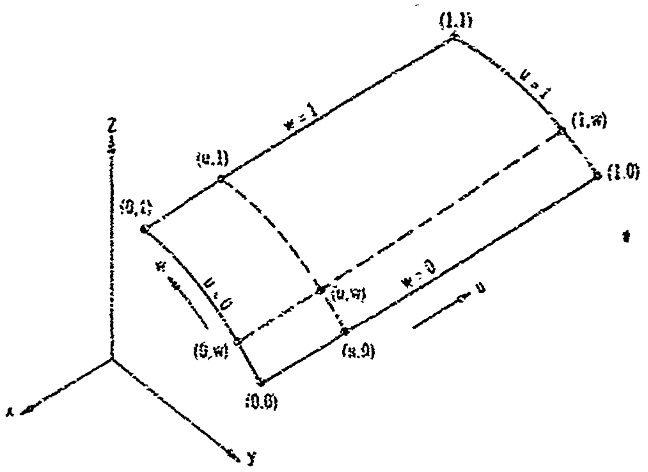
Parametric Cubic

A second technique for describing three-dimensional curved surfaces is also provided within the program. This is a mathematical surface-fit technique and is identified as the Parametric Cubic Method because of the general type of equations used.

Several different mathematical surface-fit techniques are described in the literature. The one used in this program was adopted from the formulation given by Coons of MIT (Reference 20). In this method a vehicle shape is also divided into a number of sections or patches. The size and location of each patch depends upon the shape of the surface.

The basic feature of this method is that only the surface conditions at the patch corner points are required to completely describe the surface enclosed by the boundary curves of the patch. The basic problem, however, is the determination of all the information required at these corner points, i.e., the surface equation requires corner point surface derivatives with respect to the parametric variables rather than the X, Y, Z coordinates. This has been solved by the use of additional points along the boundary curves as will be described later.

In the following discussions we will use the geometrical representation of a surface patch as illustrated in the figure below.



BOUNDARY CURVE (FOR u = 0)

$$X_i(0,w) = Aw^3 + Bw^2 + Cw + D$$

$$A = 2[X_i(0,0) - X_i(0,1)] + \frac{\partial X_i}{\partial w}(0,0) + \frac{\partial X_i}{\partial w}(0,1)$$

$$B = 3[X_i(0,1) - X_i(0,0)] - 2\frac{\partial X_i}{\partial w}(0,0) - \frac{\partial X_i}{\partial w}(0,1)$$

$$C = \frac{\partial X_i}{\partial w}(0,0) \qquad D = X_i(0,0)$$

$$\frac{\partial X_i}{\partial w} = \frac{\partial X_i}{\partial S}\frac{\partial S}{\partial w} \qquad i = 1, 2, 3 \text{ FOR } X, Y, Z$$

BLENDING FUNCTIONS

$$F_1(u) = 3u^2 - 2u^3 \qquad F_1(w) = 3w^2 - 2w^3$$

$$F_0(u) = 1 - F_1(u) \qquad F_0(w) = 1 - F_1(w)$$

SURFACE FORM

$$\begin{aligned} X_i(u,w) = {} & X_i(0,w)F_0(u) + X_i(1,w)F_1(u) + X_i(u,0)F_0(w) \\ & + X_i(u,1)F_1(w) - X_i(0,0)F_0(u)F_0(w) \\ & - X_i(0,1)F_0(u)F_1(w) - X_i(1,0)F_1(u)F_0(w) \\ & - X_i(1,1)F_1(u)F_1(w) \end{aligned}$$

Since the basic surface-fit equations and their derivatives are presented in Reference 20, they need be only reviewed briefly in this report.

The X, Y, Z coordinates of a point on the surface are related to the two parametric variables u and w. Thus, a surface in space is mapped into the u, w unit square. The basic problem is to find the position (X, Y, Z) of a point (u, w) in the interior of the section surface. The general procedure is to first find relationships for the four boundary curves. These are defined as third-order polynomials in terms of the parametric variables. The points on the boundary curves corresponding to u and w (0, w and u, 0, etc.) are then calculated. A general surface equation is used to calculate the properties at the point u, w. This equation uses blending or weighting functions to properly introduce the influence of each of the related boundary-curve points and the four corner points. The blending functions also ensure the continuity of the slopes across the boundaries between adjacent sections.

There are several methods for calculating the direction cosines of the tangent vectors required in the calculation of the corner-point derivatives. Most require the specification of additional surface-boundary points, some of which may lie on the extensions of the boundary curves. The derivatives must be calculated, since it would not be practical to measure them directly from drawings. The method in this program involves the use of circular arcs through three boundary-curve points, the middle one being a corner point.

The first step in the computational procedure is to determine the equations for the cubic boundary curves. The equation used is given by the following relationship for u = 0.

$$X_i(0, w) = Aw^3 + Bw^2 + Cw + D$$

where

$$A = 2 \left[X_i(0,0) - X_i(0,1)\right] + \frac{\partial X_i}{\partial w}(0,0) + \frac{\partial X_i}{\partial w}(0,1)$$

$$B = 3 \left[X_i(0,1) - X_i(0,0)\right] - 2\frac{\partial X_i}{\partial w}(0,0) - \frac{\partial X_i}{\partial w}(0,1)$$

$$C = \frac{\partial X_i}{\partial w}(0,0)$$

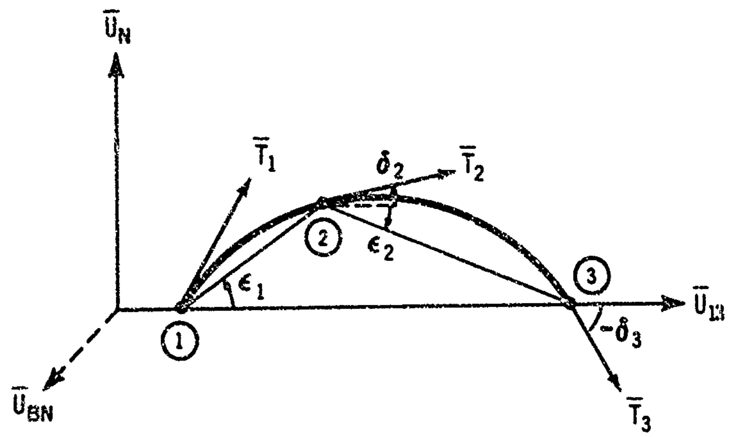
$$D = X_i(0,0)$$

Similar equations are needed for the other three boundary curves with $u = 1$, $w = 0$, and $w = 1$.

The missing items required for the solution of the above equations are the derivatives

$$\frac{\partial X_i}{\partial w}(0,0), \ \frac{\partial X_i}{\partial w}(0,1), \ \text{etc}$$

In the Arbitrary-Body Program these are determined by passing a circular arc through three points, the middle point being the corner point itself. For completeness, the development of this method is presented and the sketch below is useful in following the derivation.



This sketch is a view of the plane of the circle with $\overline{U}_{13}$ as the base coordinate. The vectors $\overline{T}_1$, $\overline{T}_2$, and $\overline{T}_3$ are tangents to the curve at the points 1, 2, and 3.

The tangents make the angles δ_1, δ_2, and δ_3 with respect to $\overline{U}_{13}$. The chord lengths make the angles ϵ_1 and ϵ_2 with respect to the vector $\overline{U}_{13}$.

One of the properties of circular arcs is that the chord angle is the average of the two tangent angles.

$$\epsilon_1 = \frac{\delta_1 + \delta_2}{2} \qquad \epsilon_2 = \frac{\delta_2 + \delta_3}{2} \qquad \epsilon_3 = \frac{\delta_1 + \delta_3}{2}$$

For the coordinate base selected ($\overline{U}_{13}$), $\epsilon_3 = 0$, therefore,

$$\delta_1 = -\delta_3 \quad \text{and} \quad \delta_2 = \epsilon_1 + \epsilon_2$$

The tangent vector at point 2 is then given by

$$\overline{T}_2 = \cos\delta_2\,\overline{U}_{13} + \sin\delta_2\overline{U}_N$$

$$\overline{U}_{13} = \frac{\overline{L}_{13}}{|\overline{L}_{13}|}$$, $\overline{L}_{13}$ is chord vector between points 1 and 3.

To determine $\overline{U}_N$, the binomial $\overline{U}_{BN}$ must first be found

$$\overline{U}_{BN} = \overline{L}_{13} \times \overline{L}_{12}$$

$$\overline{U}_{BN} = \frac{\overline{U}_{BN}}{|\overline{U}_{BN}|} \quad \text{(unit vector)}$$

$$\overline{U}_N = \overline{U}_{BN} \times \overline{U}_{13}$$

The radius vectors (X, Y, Z) for the three points are

$$\overline{r}_1 = X_1\overline{i} + Y_1\overline{j} + Z_1\overline{k}$$

$$\overline{r}_2 = X_2\overline{i} + Y_2\overline{j} + Z_2\overline{k}$$

$$\overline{r}_3 = X_3\overline{i} + Y_3\overline{j} + Z_3\overline{k}$$

The chord vectors between the points are

$$\bar{L}_{12} = \bar{r}_2 - \bar{r}_1 = (X_2 - X_1)\,\bar{i} + (Y_2 - Y_1)\,\bar{j} + (Z_2 + Z_1)\,\bar{k}$$

$$\bar{L}_{23} = \bar{r}_3 - \bar{r}_2 = (X_3 - X_2)\,\bar{i} + (Y_3 - Y_2)\,\bar{j} + (Z_3 - Z_2)\,\bar{k}$$

$$\bar{L}_{13} = \bar{r}_3 - \bar{r}_1 = (X_3 - X_1)\,\bar{i} + (Y_3 - Y_1)\,\bar{j} + (Z_3 - Z_1)\,\bar{k}$$

and the chord angles

$$\cos \epsilon_1 = \frac{\bar{L}_{12} \bullet \bar{L}_{13}}{|\bar{L}_{12}|\ |\bar{L}_{13}|} \qquad \cos \epsilon_2 = \frac{\bar{L}_{23} \bullet \bar{L}_{13}}{|\bar{L}_{23}|\ |\bar{L}_{13}|}$$

For convenience we will use the shortened notation:

$$L_{12} = \left|\bar{L}_{12}\right|, \text{ etc.}$$

$$\bar{U}_{13} = \left(\frac{X_3 - X_1}{L_{13}}\right)\bar{i} + \left(\frac{Y_3 - Y_1}{L_{13}}\right)\bar{j} + \left(\frac{Z_3 - Z_1}{L_{13}}\right)\bar{k}$$

$$= \ell_1\,\bar{i} + m_1\,\bar{j} + n_1\,\bar{k}$$

Similarly

$$\bar{U}_{12} = \ell_2\,\bar{i} + m_2\,\bar{j} + n_2\,\bar{k}$$

$$\bar{U}_{BN} = \frac{\bar{U}_{BN}}{|\bar{U}_{BN}|} = \bar{U}_{13} \times \bar{U}_{12} = \begin{vmatrix} i & j & k \\ \ell_1 & m_1 & n_1 \\ \ell_2 & m_2 & n_2 \end{vmatrix}$$

$$= (m_1\,n_2 - m_2\,n_1)\,\bar{i} - (\ell_1\,n_2 - \ell_2\,n_1)\,\bar{j} + (\ell_1\,m_2 - \ell_2\,m_1)\,\bar{k}$$

$$\bar{U}_N = \bar{U}_{BN} \times \bar{U}_{13} = \begin{vmatrix} i & j & k \\ (\ ) & -(\ ) & (\ ) \\ \ell_1 & m_1 & n_1 \end{vmatrix}$$

$$= \left[- n_1 ({}'_1 n_2 - \ell_2 n_1) - m_1 (\ell_1 m_2 - \ell_2 m_1)\right] i$$

$$- \left[n_1 (m_1 n_2 - m_2 n_1) - \ell_1 (\ell_1 m_2 - \ell_2 m_1)\right] j$$

$$+ \left[m_1 (m_1 v_2 - m_2 n_1) + \ell_1 (\ell_1 n_2 - \ell_2 n_1)\right] k$$

$$\overline{U}_N = \ell_N \bar{i} + m_N \bar{j} + n_N \bar{k}$$

And finally we obtain the tangent vector

$$\overline{T}_2 = (\ell_1 \cos\delta + \ell_N \sin\delta)\,\bar{i} + (m_1 \cos\delta + m_N \sin\delta)\,\bar{j}$$

$$= (n_1 \cos\delta + n_N \sin\delta)\,\bar{k}$$

where

$$\ell_1 = \frac{X_3 - X_1}{L_{13}}, \qquad m_1 = \frac{Y_3 - Y_1}{L_{13}}, \qquad n_1 = \frac{Z_3 - Z_1}{L_{13}}$$

$$L_{13} = \left[(X_2 - X_1)^2 + (Y_3 - Y_1)^2 + (Z_3 - Z_1)^2\right]^{1/2}$$

$$\ell_N = -\left[n_1 (\ell_1 n_2 - \ell_2 n_1) + m_1 (\ell_1 m_2 - \ell_2 m_1)\right]$$

$$m_N = -\left[n_1 (m_1 n_2 - m_2 n_1) + \ell_1 (\ell_1 m_2 - \ell_2 m_1)\right]$$

$$n_N = \left[m_1 (m_1 n_2 - m_2 n_1) + \ell_1 (\ell_1 n_2 - \ell_2 n_1)\right]$$

and

$$\ell_2 = \frac{X_2 - X_1}{L_{12}}, \qquad m_2 = \frac{Y_2 - Y_1}{L_{12}}, \qquad n_2 = \frac{Z_2 - Z_1}{L_{12}}$$

$$L_{12} = \left[(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2\right]^{1/2}$$

The final end point derivatives are then found from

$$\frac{\partial X_i}{\partial w} = T_i \frac{\partial S}{\partial w}$$

where

$$\frac{\partial s}{\partial w} = \text{the boundary length since } \Delta w = 1 \text{ on the unit square patch}$$

$$\Delta S = \sum_{I=2}^{I=NB-1} \left[(X_{I+1} - X_I)^2 + (Y_{I+1} - Y_I)^2 + (Z_{I+1} - Z_I)^2 \right]^{1/2}$$

I = 2 at the starting corner point

I = NB - 1 at the final point on the boundary curve

NB = number of points input on the boundary with one point extending off each end of the boundary curve.

Once the boundary curves are found the values required for the general surface equation can be calculated. This equation is given below.

$$\begin{aligned} X_i(u,w) = {} & X_i(0,w)F_o(u) + X_i(1,w)F_i(u) + X_i(u,0)F_o(w) \\ & + X_i(u,1)F_1(w) - X_i(0,0)F_o(u)F_o(w) \\ & - X_i(0,1)F_o(u)F_1(w) - X_i(1,0)F_1(u)F_o(w) \\ & - X_i(1,1)F_1(u)F_1(w) \end{aligned}$$

where the terms F_o and F_1 are blending functions given by

$$F_1(u) = 3u^2 - 2u^3 \qquad F_1(u) = 3w^2 - 2w^3$$

$$F_o(u) = 1 - F_1(u) \qquad F_o(w) = 1 - F_1(w)$$

The program does not use the parametric cubic geometry data directly in the pressure calculations. Instead, the parametric cubic data are used in creating surface elements by a systematic variation of the parametric variables w, and u.

One advantage of the mathematical surface-fit technique over the plane-distributed-element method is the smaller number of surface points required to describe a shape. However, additional points are required on the boundaries to determine the required corner derivatives. This method is not as adaptable to semiautomatic data-reading techniques, since the organization of the required input data is more complex. The accuracy of this method depends upon the distribution and orientation of the surface sections, just as the plane-distributed-element method depends upon the distribution of the elements.

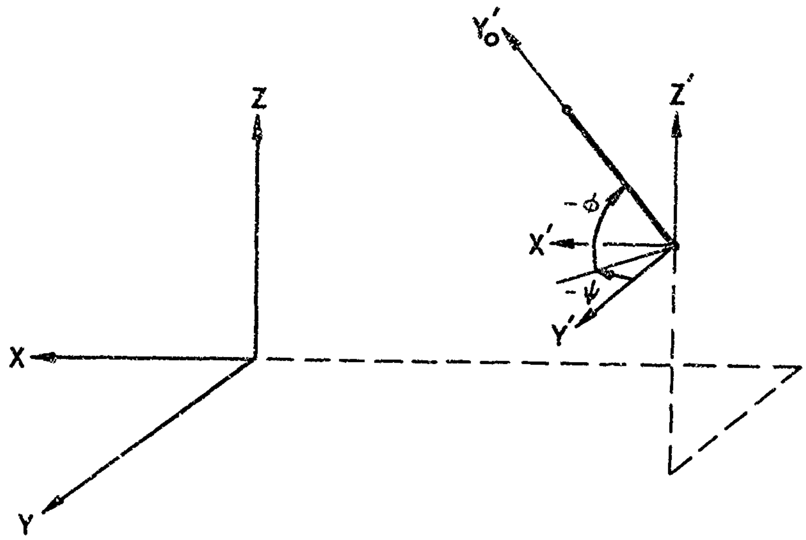
Auxiliary Geometry Methods

The program has two other geometry generation options. These are (1) the ellipse surface generation, and (2) the slab delta geometry generation. In each of these methods the program calculates the necessary Y-Z coordinate data at specified X stations. In the ellipse generation simple ellipse equations are used; in the slab delta geometry generation, similar equations are used along with the necessary equations for the bottom and top parts inboard of the nose and leading edge.

Control Surface Geometry

The geometry data for a control surface flap are input to the program in the undeflected position. The methods used in transforming these data to the required deflected position are outlined in the following discussion.

The coordinate system used in these derivations is shown in the drawing below.



The general procedure involves a coordinate shift and an appropriate rotation to a hinge-line centered coordinate system such that the new Y-axis (Y_0') lies along the hinge line. For ψ and ϕ equal to zero and with the flap surface normal in the negative z-direction, the hinge-line centered coordinate system has the same directions as the body-axis system. The corner points, centroid, and normal vector (direction cosines) for each element of the flap are transformed into this system. Since the flap is a rigid body this information is independent of flap deflection and the hinge moment factor (moment per unit normal force) need only be determined once. However, the force magnitude is a function of the deflection angle and requires having the geometry of the deflected flap in the vehicle-centered coordinates.

The coordinate system shift is given by

$$X' = X - X_{HL_4}$$

$$Y' = Y - Y_{HL_4}$$

$$Z' = Z - Z_{HL_4}$$

Where

$(\ \ )_{HL_4}$ is to point 4 on the hinge line

The new coordinates of the flap in the shifted and transformed coordinate system are given by

$$\begin{bmatrix} X'_0 \\ Y'_0 \\ Z'_0 \end{bmatrix} = \begin{bmatrix} E \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}$$

Where

$$\begin{bmatrix} E \end{bmatrix} = \begin{bmatrix} \Phi \end{bmatrix} \begin{bmatrix} \Psi \end{bmatrix}$$

$$\begin{bmatrix} \Psi \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ψ = rotation about the Z'-axis

$$\begin{bmatrix} \Phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

ϕ = rotation about the X'_o-axis

The final rotation to the deflected position (δ_e is the control surface deflection) is given by

$$\begin{bmatrix} X'_{0\delta_e} \\ Y'_{0\delta_e} \\ Z'_{0\delta_e} \end{bmatrix} = \begin{bmatrix} \delta_e \end{bmatrix} \begin{bmatrix} X'_0 \\ Y'_0 \\ Z'_0 \end{bmatrix} = \begin{bmatrix} \delta_e \end{bmatrix} \begin{bmatrix} E \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z \end{bmatrix}$$

where

$$\begin{bmatrix} \delta_e \end{bmatrix} = \begin{bmatrix} \cos\delta_e & 0 & -\sin\delta_e \\ 0 & 1 & 0 \\ \sin\delta_e & 0 & \cos\delta_e \end{bmatrix}$$

The coordinates of the deflected flap are then transformed back to vehicle centered coordinate system, first through the inverse rotation

$$\begin{bmatrix} X'_{\delta_e} \\ Y'_{\delta_e} \\ Z'_{\delta_e} \end{bmatrix} = \begin{bmatrix} E \end{bmatrix}^{-1} \begin{bmatrix} X'_{0\delta_e} \\ Y'_{0\delta_e} \\ Z'_{0\delta_e} \end{bmatrix}$$

and then by the coordinate shift

$$X_{\delta_e} = X'_{\delta_e} + X_{HL_4}$$
$$Y_{\delta_e} = Y'_{\delta_e} + Y_{HL_4}$$
$$Z_{\delta_e} = Z'_{\delta_e} + Z_{HL_4}$$

The rotation angles are defined for a right-handed system and are found from the relationships

$$\psi = \sin^{-1}\left(\frac{X_{HL_1} - X_{HL_4}}{L_{XY}}\right) \text{ and } \phi = -\sin^{-1}\left(\frac{Z_{HL_1} - Z_{HL_4}}{L_{YZ}}\right)$$

where

$$L_{XY} = \left[(X_{HL_1} - X_{HL_4})^2 + (Y_{HL_4})^2\right]^{1/2}$$

and

$$L_{YZ} = \left[L_{XY}^2 + (Z_{HL_1} - Z_{HL_4})^2\right]^{1/2}$$

A check is made in the program and if $Y_{HL_1} < Y_{HL_4}$ then the yaw rotation angle is set to $\psi = \pi - \psi$ to position the hinge line in the proper quadrant.

The third rotation angle δ_e is, of course, specified for a given problem. It should be noted in the present approach, that the coordinate system is rotated through the angle δ_e, positive in the right-handed sense for the system defined. Relative to the physical problem, positive δ_e corresponds to a flap deflection into the flow.

The hinge moment factor (HMFCT) is simply a function of the element geometry and location, and is defined as follows. The total moment of an element is (considering only inviscid forces)

$$\bar{M}'_0 = -(\bar{R}'_0 \times \bar{F}) = P(\bar{R}'_0 \times \bar{N}'_0)\,\mathrm{AREA}$$

where

$\bar{R}'_0$ is the radius vector to the element centroid,

P is the net surface pressure,

and AREA is the element area.

The hinge line moment is just the $\bar{Y}'_0$-component of the total moment;

$$M_{HL} = M_{Y'_0} = \bar{j}'_0 \cdot \bar{M}'_0 = P\,(\mathrm{HMFCT})$$

where

$$\mathrm{HMFCT} = (Z'_0 N_{X'_0} - X'_0 N_{Z'_0})\,\mathrm{AREA}$$

Once the deflected flap is properly oriented in the vehicle centered coordinates, the force on each element and hinge moment are determined.

Graphics -- Picture Drawing Program

The perspective drawing computer program is an important component of the Arbitrary-Body Hypersonic Force Analysis System. Its use in this system is in providing graphical perspective drawings of the geometric description input to the arbitrary body force program.* The purpose of these drawings is to allow the engineer to detect errors in the geometric input data to the arbitrary body force program.

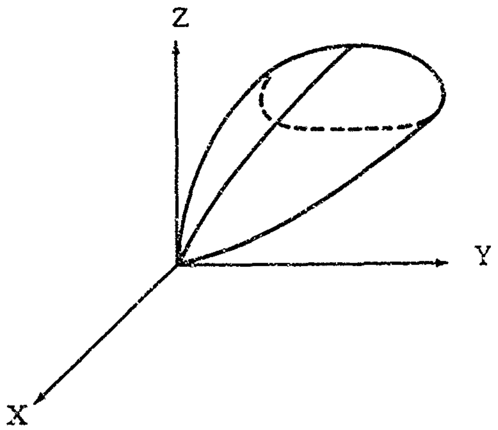
As explained previously, the body shape is defined by input sets of points in three-dimensional space. A grouping of four surface points is used to describe a surface element. An organization of a large number of related elements forms a body section and a number of sections may be used to give a complete description of the shape. The equations required to produce the perspective pictures are derived in the following paragraphs.

*The drawings made by this program are not true perspective drawings but are a limiting case where the vanishing points have been moved to infinity.

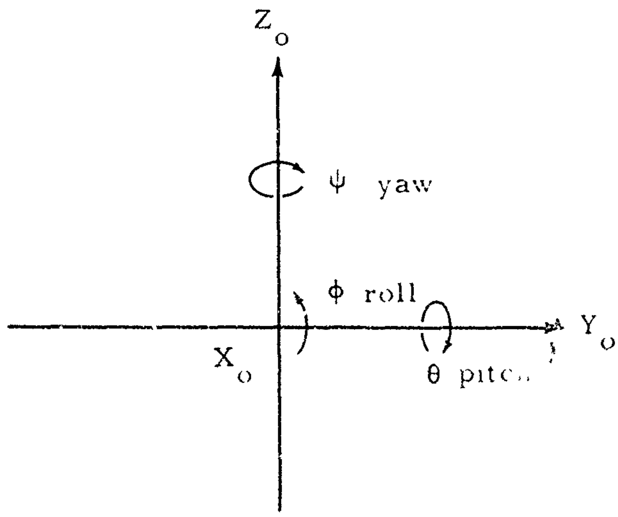
Each point on the surface is described by its coordinates in the body reference coordinate system.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The body reference coordinate system is assumed to be a conventional right-handed Cartesian system as illustrated below.



To create the perspective drawings illustrated in this report each surface point on the body must be rotated to the desired viewing angle and then transformed into a coordinate system in the plane of the paper. With zero rotation angles the body coordinate system is coincident with the fixed system in the plane of the paper.



The rotations of the body and its coordinate system to give a desired viewing angle are specified by a yaw-pitch-roll sequence (ψ, θ, ϕ). This rotation is given by the following relationship:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \phi \end{bmatrix} \begin{bmatrix} \theta \end{bmatrix} \begin{bmatrix} \psi \end{bmatrix} \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix}$$

Where the rotation matrices are

$$\begin{bmatrix} \psi \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\begin{bmatrix} \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

or

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} E \end{bmatrix} \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix}$$

where $\begin{bmatrix} E \end{bmatrix} = \begin{bmatrix} \phi \end{bmatrix} \begin{bmatrix} \theta \end{bmatrix} \begin{bmatrix} \psi \end{bmatrix}$

Since each point on the surface is given by its coordinates in the X, Y, Z system, its position in the fixed coordinate system (X_o, Y_o, Z_o) may be found by reversing the above process.

$$\begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} = \begin{bmatrix} E \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

If we carry out this operation we obtain

$$\begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & -\sin\psi\cos\phi+\sin\theta\cos\psi\sin\phi & \sin\psi\sin\phi+\sin\theta\cos\psi\cos\phi \\ \cos\theta\sin\psi & \cos\psi\cos\phi+\sin\theta\sin\psi\sin\phi & -\cos\psi\sin\phi+\sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$X_o = X(\cos\theta\cos\psi) + Y(-\sin\psi\cos\phi+\sin\theta\cos\psi\sin\phi) + Z(\sin\psi\sin\phi+\sin\theta\cos\psi\cos\phi)$$

$$Y_o = X(\cos\theta\sin\psi) + Y(\cos\psi\cos\phi+\sin\theta\sin\psi\sin\phi) + Z(-\cos\psi\sin\phi+\sin\theta\sin\psi\cos\phi)$$

$$Z_o = X(-\sin\theta) + Y(\cos\theta\sin\phi) + Z(\cos\theta\cos\phi)$$

We may now use these last two equations to transform a given point on the body (X, Y, Z) with a specified set of rotation angles (ψ, ϕ, θ) into the plane of the paper (the Y_o, Z_o system). With the SC-4020 library subroutines it now becomes a simple matter to plot these data and to connect the related points with straight lines.

In the surface fit technique used in this program and described in Reference 22, each input element is replaced by a plane quadrilateral surface element whose characteristics are used for all subsequent calculations. These characteristics include the area, centroid, and the direction cosines of the surface unit normal. The surface unit normals may be transformed through the required rotation angles just as was done for the individual points. The resulting value of the component of the unit normal in the X_o direction (out of the plane of the paper) may be found from the following equation.

$$n_{x_o} = n_x(\cos\theta\cos\psi)+n_y(-\sin\psi\cos\phi+\sin\theta\cos\psi\sin\phi)+n_z(\sin\psi\sin\phi+\sin\theta\cos\psi\cos\phi)$$

where n_x, n_y, n_z are the components of the surface unit normal in the vehicle reference system.

If n_{x_o} is positive then the surface element is facing the viewer. If n_{x_o} is negative the element faces away from the plane of the paper. This result is used in the program to provide the capability of deleting most of those elements on a vehicle that normally could not be seen by a viewer. The resulting picture is thus made more realistic and confusing elements which are on the back side of the vehicle do not appear. No criterion is provided, however, for the deletion of those elements that face the viewer but are blocked by other body components. This may be accomplished by a proper selection of viewing angle or by a physical deletion of the offending section from the input data.

COMPUTATION OF VEHICLE FORCES

Calculation of Local Flow Conditions

In the preceding derivations we have converted the input element into a plane quadrilateral element. The quadrilateral is described by its area, the coordinates of the centroid of the element, and by the direction cosines of the surface unit normal. In the force calculation methods we must also know the angle that the element makes with the free-stream velocity vector (the impact angle). This angle changes as

the vehicle attitude (angle of attack and yaw angle) changes. The impact angle may be found from the following relationship:

$$\delta = \pi/2 - \theta$$

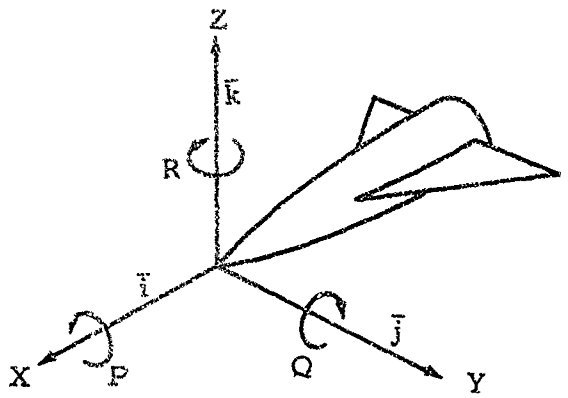
$$\cos \theta = \frac{\bar{n} \bullet \bar{V}}{|\bar{n}|\ |\bar{V}|}$$

where

$\bar{n}$ is the unit normal outward from the surface with direct on cosines n_x , n_y, n_z

$\bar{V}$ is the local velocity vector with direction cosines in the vehicle coordinate system given by V_x , V_y , V_z

The direction cosines of the unit surface normal are given by the quadrilateral calculations. The value of the local velocity vector V depends upon the vehicle attitude with respect to the free-stream direction and its angular rotation rates, and is derived in the discussion below. The rotation directions are consistent with the conventional stability body-axis system. The coordinate system, however, is changed to be consistent with the geometric description system discussed previously.



where

P = rolling velocity
Q = pitching velocity
R = yawing velocity
Ω = total angular velocity

The movement of a given element of the vehicle with respect to the free-stream depends upon the vehicle rotation rate and the position of the element relative to the rotation center. The radius vector from an arbitrary reference point on the vehicle to a point on the surface is given by

$$\bar{r} = (x - x_o)\ \bar{i} + (y - y_o)\ \bar{j} + (z - z_o)\ \bar{k}$$

where x_o, y_o, z_o is the moment reference point (center of gravity).

The total angular velocity is given by

$$\bar{\Omega} = P\,\bar{i} - Q\,\bar{j} - R\,\bar{k}$$

The free-stream velocity vector is given by

$$\bar{V}_\infty = V_{\infty_x}\,\bar{i} + V_{\infty_y}\,\bar{j} + V_{\infty_z}\,\bar{k}$$

The free-stream velocity components are given by the following relationships for a conventional yaw-pitch sequence.

$$V_{\infty_x} = -V_\infty \cos\alpha \cos\beta$$

$$V_{\infty_y} = V_\infty \sin\beta$$

$$V_{\infty_z} = V_\infty \cos\beta \sin\alpha$$

where

α = angle of attack
β = sideslip angle

The total velocity vector relative to the surface element is obtained by combining the above relationships as follows:

$$\bar{V} = \bar{V}_\infty - \bar{\Omega} \times \bar{r}$$

The local velocity vector therefore becomes

$$\begin{aligned}\bar{V} = {} & \left\{V_{\infty_x} + \left[Q(z-z_o) - R(y-y_o)\right]\right\}\bar{i} \\ & + \left\{V_{\infty_y} + \left[R(x-x_o) + P(z-z_o)\right]\right\}\bar{j} \\ & + \left\{V_{\infty_z} - \left[P(y-y_o) + Q(x-x_o)\right]\right\}\bar{k}\end{aligned}$$

or

$$\bar{V} = V_x\,\bar{i} + V_y\,\bar{j} + V_z\,\bar{k}$$

where

$$V_x = V_{\infty_x} + \left[Q(z-z_o) - R(y-y_o)\right]$$

$$V_y = V_{\infty_y} + \left[R(x-x_o) + P(z-z_o)\right]$$

$$V_z = V_{\infty_z} - \left[P(y-y_o) + Q(x-x_o)\right]$$

The total local velocity is given by

$$V_{local} = \sqrt{V_x^2 + V_y^2 + V_z^2}$$

The conventional surface impact angle is then given by

$$\delta = \pi/2 - \cos^{-1}\left(\frac{-n_x V_x - n_y V_y - n_z V_z}{V_{local}}\right)$$

where

n_x, n_y, n_z are the outward surface unit normal direction cosines

Vehicle Coefficients and Derivatives

In the program force calculations the pressure on each element is calculated completely independent of all other elements (except the shock-expansion method). If the vehicle is rotating the local pressure coefficient must be corrected back to free stream conditions. This is accomplished by the following relationship.

$$C_p = C_{p_{local}} \left(\frac{V_{local}}{V_\infty}\right)^2$$

In the arbitrary body force calculation program the pressure coefficient on each element is calculated completely independent of all other elements. The contributions of all the elements are then summed to give the total force components. The basic relationships to accomplish this are given below:

axial force $C_A = \frac{1}{S_{ref}} \sum C_p \, n_x \Delta A$

side force $C_Y = \frac{1}{S_{ref}} \sum C_p \, n_y \Delta A$

normal force $C_N = -\frac{1}{S_{ref}} \sum C_p \, n_z \Delta A$

where

ΔA = element area

Note: The minus sign is needed because of the sign convention on Z in the body coordinate system, directed upward.

The moment coefficients are obtained by a summation of the component forces multiplied by the distance from the element centroid to the reference moment center.

$$C_l = \sum(C_Y \frac{z}{b}) + \sum(C_N \frac{y}{b})$$ rolling moment

$$C_m = \sum(C_N \frac{x}{\bar{c}}) + \sum(C_A \frac{z}{\bar{c}})$$ pitching moment

$$C_n = \sum(C_Y \frac{x}{b}) - \sum(C_A \frac{y}{b})$$ yawing moment

where

b	=	reference span (lateral and directional moment coefficient reference length)
$\bar{c}$	=	mean aerodynamic chord (for longitudinal moment reference)
x, y, z	=	distances from the center of gravity
	=	$x_{centroid} - x_{cg}$, etc.

The conversion of the axial force and normal force coefficients to lift and drag coefficients requires the following rotation matrices.

$$\begin{bmatrix} C_D \\ C_Y' \\ C_L \end{bmatrix} = \left[E\right]^{-1} \begin{bmatrix} C_A \\ C_Y \\ C_N \end{bmatrix}$$

where

$$\left[E\right]^{-1} = \begin{bmatrix} \cos\alpha \cos\beta & -\sin\beta & \sin\alpha \cos\beta \\ \cos\alpha \sin\beta & \cos\beta & \sin\alpha \sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

α = angle of attack (+ nose up)
β = sideslip angle (+ nose left)

$$C_D = C_A \cos\alpha \cos\beta - C_Y \sin\beta + C_N \sin\alpha \cos\beta$$

$$C_Y' = C_A \cos\alpha \sin\beta + C_Y \cos\beta + C_N \sin\alpha \sin\beta$$

$$C_L = -C_A \sin\alpha + C_N \cos\alpha$$

The vehicle static stability derivatives in angle of attack and sideslip are calculated by the method of small perturbations as indicated below:

$$C_{A_\alpha} = \frac{(C_A)_{\alpha+\Delta\alpha} - (C_A)_\alpha}{\Delta\alpha}$$

$$C_{N_\alpha} = \frac{(C_N)_{\alpha+\Delta\alpha} - (C_N)_\alpha}{\Delta\alpha}$$

$$C_{m_\alpha} = \frac{(C_m)_{\alpha+\Delta\alpha} - (C_m)_\alpha}{\Delta\alpha}$$

$$C_{Y_\beta} = \frac{(C_Y)_{\beta+\Delta\beta} - (C_Y)_\beta}{\Delta\beta}$$

$$C_{n_\beta} = \frac{(C_n)_{\beta+\Delta\beta} - (C_n)_\beta}{\Delta\beta}$$

The damping derivatives due to vehicle rotation rate are given in a similar manner

$$C_{m_q} = \left[\frac{(C_m)_{q+\Delta q} - (C_m)_q}{\Delta q}\right] \Big/ \frac{\bar{c}}{2V}$$

etc.

The control surface derivatives are also calculated by the method of small perturbations.

$$C_{L_\delta} = \frac{(C_L)_{\delta+\Delta\delta} - (C_L)_\delta}{\Delta\delta}$$

$$C_{m_\delta} = \frac{(C_m)_{\delta+\Delta\delta} - (C_m)_\delta}{\Delta\delta}$$

$$C_{\ell_\delta} = \frac{(C_\ell)_{\delta+\Delta\delta} - (C_\ell)_\delta}{\Delta\delta}$$

$$C_{Y_\delta} = \frac{(C_Y)_{\delta+\Delta\delta} - (C_Y)_\delta}{\Delta\delta}$$

$$C_{n_\delta} = \frac{\left(C_n\right)_{\delta + \Delta\delta} - \left(C_n\right)_\delta}{\Delta\delta}$$

$$C_{N_\delta} = \frac{\left(C_N\right)_{\delta + \Delta\delta} - \left(C_N\right)_\delta}{\Delta\delta}$$

Inviscid Force Calculation Methods

Many of the pressure calculation methods used in the analysis of high-speed shapes are listed in Figure 2 . An attempt has been made in the preparation of this figure to indicate the interrelationships of the methods (the information can, of course, be organized in many different ways). Some of these methods are more applicable to the arbitrary-body problem than others.

The method of characteristics is the eventual ideal approach for the calculation of forces on three-dimensional shapes at high speeds. It will require starting solutions for three-dimensional blunt bodies of arbitrary shape. The development of a method of calculating three-dimensional boundary layers would permit the use of an iterative process to account for the viscous-inviscid interaction. Although this approach has been used for some very simple shapes, the complete solution for arbitrary shapes is some time away. Significant progress is being made in the solution of the inviscid flow field by the method of characteristics. However, present mathematical techniques and digital-computer size and speed capability prevent application to typical preliminary-design problems. Application must be reserved for simple shapes or important detail design applications where very large computer times might be acceptable.

Many of the other methods shown in Figure 2 would be useful force-calculation methods for inclusion in an arbitrary-body system. The selection of the proper method in a given application depends upon the vehicle-component shape and flight condition and must be selected by the engineer on the basis of his knowledge and experience in the use of each method.

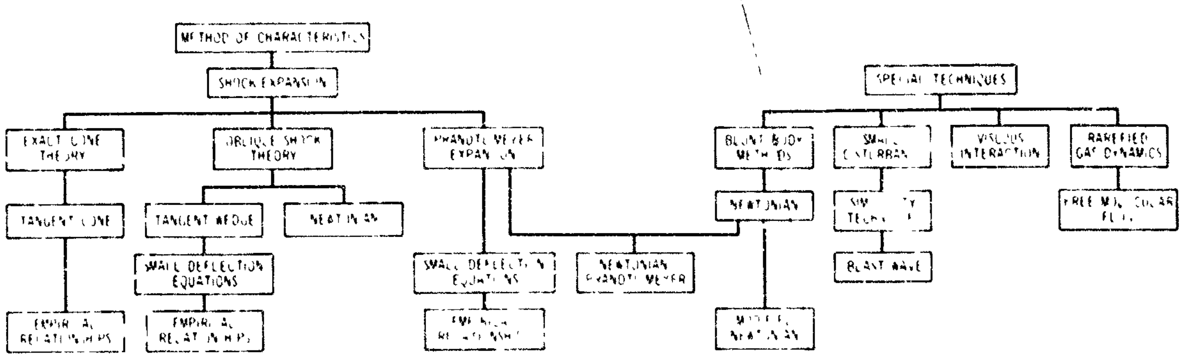


Figure 2. Pressure Calculation Methods

The arbitrary body force computer program contains a number of optional methods for calculating the pressure coefficient. In each method the only geometric parameter required is the element impact angle, δ, or the change in the angle of an element from a previous point.

Before the program calculates the pressure on each surface element, it checks to see if the element is facing the flow (in an impact region) or facing away from the flow (in a shadow region). The methods to be used in calculating the pressure in impact and shadow regions may be specified independently. A summary of the program pressure options is presented below.

PRESSURE CALCULATION METHODS – MARK III MOD 0 PROGRAM

Impact Flow	Shadow Flow
1. Modified Newtonian	1. Newtonian ($C_p = 0$)
2. Modified Newtonian+Prandtl-Meyer	2. Modified Newtonian+Prandtl-Meyer
3. Tangent wedge	3. Prandtl-Meyer from free-stream
4. Tangent-wedge empirical	4. OSU blunt body empirical
5. Tangent-cone empirical	5. Van Dyke Unified
6. OSU blunt body empirical	6. High Mach base pressure
7. Van Dyke Unified	7. Shock-expansion
8. Blunt-body shear force	8. Input pressure coefficient
9. Shock-expansion	9. Free molecular flow
10. Free molecular flow	10. Boundary-layer induced pressures
11. Input pressure coefficient	
12. Hankey flat-surface empirical	
13. Delta wing empirical	
14. Dahlem-Buck empirical	
15. Blast wave	
16. Modified tangent-cone	
17. Boundary layer induced pressures	

Since most of these methods are adequately discussed in the literature they will be reviewed only briefly in this document. The blunt-body shear force and the boundary-layer induced pressure methods are discussed in detail in the section describing Viscous Force Methods.

Modified Newtonian

This method is probably the most widely used of all the hypersonic force analysis techniques. The major reason for this is its simplicity. Like all the force calculation methods, however, its validity in any particular application depends upon the flight condition and the shape of the vehicle or component being considered. Its most general application is for blunt shapes at high hypersonic speed. The usual form of the modified Newtonian pressure coefficient is

$$C_p = K \sin^2\delta$$

In true Newtonian flow ($M = \infty$, $\gamma = 1$) the parameter K is taken as 2. In the various forms of modified Newtonian theory, K is given values other than 2 depending on the type of modified Newtonian theory used. K is frequently taken as being equal to the stagnation pressure coefficient. In other forms it is determined by the following relationship (Reference 23).

$$K = \frac{C_{p_{nose}}}{\sin^2 \delta_{nose}}$$

where

$C_{p_{nose}}$ = the exact value of the pressure coefficient at the nose or leading edge

δ_{nose} = impact angle at the nose or leading edge

In other work K is determined purely on an empirical basis.

$$K = \text{fn}\,(M, \alpha, \text{shape})$$

When modified Newtonian theory is used, the pressure coefficient in shadow regions (δ is negative) is usually set equal to zero.

Modified Newtonian Plus Prandtl-Meyer

This method, described as the blunt body Newtonian + Prandtl-Meyer technique, is based on the analysis presented by Kaufman in Reference 24. The flow model used in this method assumes a blunt body with a detached shock, followed by an expansion around the body to supersonic conditions. This method uses a combination of modified Newtonian and Prandtl-Meyer expansion theory. Modified Newtonian theory is used along the body until a point is reached where both the pressure and the pressure gradients match those that would be calculated by a continuing Prandtl-Meyer expansion.

The calculation procedure derived for determining the pressure coefficient using the blunt body Newtonian + Prandtl-Meyer technique is outlined below.

1. Calculate free-stream static to stagnation pressure ratio

$$P = \frac{p_\infty}{p_o} = \left[\frac{2}{(\gamma + 1)\, M_\infty^2}\right]^{\frac{\gamma}{\gamma - 1}} \left[\frac{2\gamma M_\infty^2 - (\gamma - 1)}{\gamma + 1}\right]^{\frac{1}{\gamma - 1}}$$

2. Assume a starting value of the matching Mach number, M_q (for $\gamma = 1.4$ assume $M_q = 1.35$)

3. Calculate matching point to free-stream static pressure ratio

$$Q = \frac{p_q}{p_o} = \left[\frac{2}{2 + (\gamma - 1) M_q^2}\right]^{\frac{\gamma}{\gamma - 1}}$$

4. Calculate new free-stream static to stagnation pressure ratio

$$P_c = Q \left[1 - \frac{\gamma^2 M_q^4 Q}{4(M_q^2 - 1)(1 - Q)}\right]$$

5. Assume a new matching point Mach number (1.75) and repeat the above steps to obtain a second set of data.

6. With the above two tries use a linear interpolation equation to estimate a new matching point Mach number. This process is repeated until the solution converges.

7. Calculate the surface slope at the matching point

$$\sin^2 \delta_q = \frac{Q - P}{1 - P}$$

8. Use the Prandtl-Meyer expansion equations to find the Mach number on the surface element, M_δ

9. Calculate the surface pressure ratio

$$\frac{p_\delta}{p_o} = \eta_c \left[1 + \frac{\gamma - 1}{2} M_\delta^2\right]^{-\frac{\gamma}{\gamma - 1}}$$

where

η_c is provided as an empirical correction factor

p_δ is the pressure on the element of interest

10. Calculate the surface to free-stream pressure ratio

$$\frac{p_\delta}{p_\infty} = \left(\frac{1}{P}\right) \left(\frac{p_\delta}{p_o}\right)$$

11. Calculate the surface pressure coefficient

$$C_{P_\delta} = \frac{2}{\gamma M_\infty^2}\left(\frac{p_\delta}{p_\infty} - 1\right)$$

The results of typical calculations using the above procedure are shown in Figure 3. Note that the calculations give a positive pressure coefficient at a zero impact angle. As pointed out in several references these results correlate well with test data for blunt shapes. However, if the surface curvature changes gradually to zero slope some distance from the blunt stagnation point the pressure calculated by this method will be too high. This is caused by characteristics near the nose intersecting the curved shock system and being reflected back onto the body. If the zero slope is reached near the nose (such as in a hemisphere or a cylinder) this effect has not had time to occur.

Tangent-Wedge

The tangent-wedge and tangent-cone theories are frequently used to calculate the pressures on two-dimensional bodies and bodies of revolution, respectively. These methods are really empirical in nature since they have no firm theoretical basis. They are suggested, however, by the results of more exact theories that show that the pressure on a surface in impact flow is primarily a function of the local impact angle. In this program the tangent-wedge pressures are calculated using the oblique shock relationships of NACA TR-1135 (Reference 25). The basic equation used is the cubic given by

$$\left(\sin^2\theta_s\right)^3 + b\left(\sin^2\theta_s\right)^2 + c\left(\sin^2\theta_s\right) + d = 0 \qquad \text{or}$$

$$R^3 + b\,R^2 + c\,R + d = 0$$

where

θ_s = shock angle

δ = wedge angle

$$b = -\frac{M^2 + 2}{M^2} - \gamma \sin^2\delta$$

$$c = \frac{2M^2 + 1}{M^4} + \left[\frac{(\gamma + 1)^2}{4} + \frac{\gamma - 1}{M^2}\right]\sin^2\delta$$

$$d = -\frac{\cos^2\delta}{M^4}$$

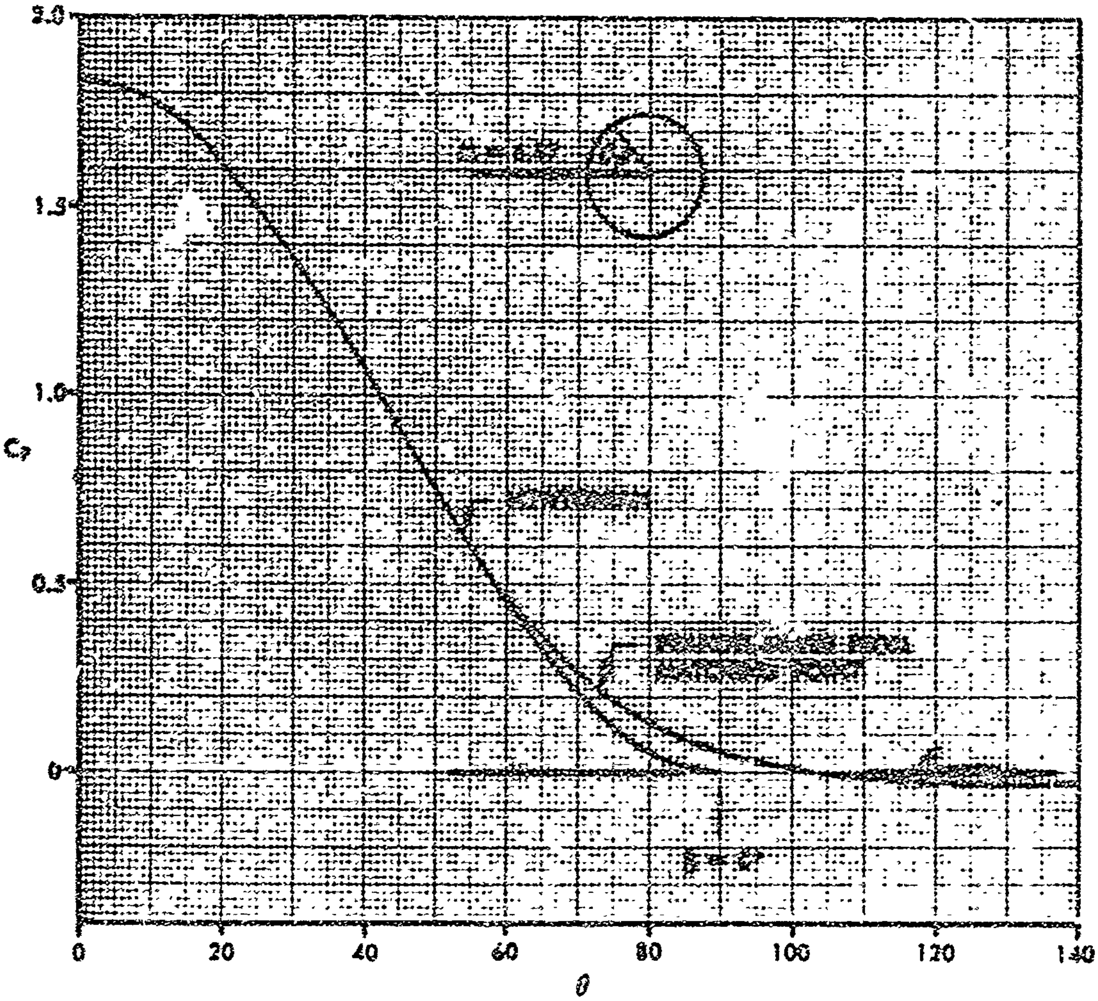


Figure 3. Blunt Body Newtonian + Prandtl-Meyer Pressure Results

The roots of the above cubic equation may be obtained by using the trigonometric solution procedure (see Reference 26) as indicated below.

$$y_1 \quad = \quad 2\ \sqrt{-\,p/3}\ \cos\ (\omega/3)\ -\ b/3$$

$$y_2 \quad = \quad -2\ \sqrt{-\,p/3}\ \cos\ (\omega/3 + 60°)\ -\ b/3$$

$$y_3 \quad = \quad -2\ \sqrt{-\,p/3}\ \cos\ (\omega/3 - 60°)\ -\ b/3$$

$$R_1 \quad = \quad y_1 - b/3$$

$$R_2 \quad = \quad y_2 - b/3$$

$$R_3 \quad = \quad y_3 - b/3$$

where

y_i = roots of the reduced cubic equation

$$p \quad = \quad -\frac{b^2}{3} + c$$

$$q \quad = \quad 2(b/3)^3 - \frac{bc}{3} + d$$

$$\cos\omega \quad = \quad -\frac{q}{2\sqrt{-(p/3)^3}}$$

R_i = $\sin^2\theta_s$ = roots of the cubic equation

The smallest of the three roots corresponds to a decrease in entropy and is disregarded. The largest root is also disregarded since it never appears in physical actuality.

For small deflections, the cubic solution becomes very sensitive to numerical accuracy; that is, to the number of significant digits carried. Since this is dependent on the particular machine employed, an alternate procedure is used.

When the flow deflection angle is equal to or less than 2.0 degrees, the following equation is used instead of the above cubic relationships (Reference 27):

$$\sin^2 \theta_s = \frac{1}{M^2} + \frac{\gamma + 1}{2} \frac{\delta}{\sqrt{M^2 - 1}}$$

Once the shock angle is obtained the remaining flow properties may be found from the relationships of Reference 25.

$$\text{density} = \rho_2 = \rho \left[\frac{6\, M^2 \sin^2 \theta_s}{M^2 \sin^2 \theta_s + 5} \right]$$

$$\text{temperature} = T_2 = T \left[\frac{7(M^2 \sin^2 \theta_s - 1)\,(M^2 \sin^2 \theta_s + 5)}{36\, M^2 \sin^2 \theta_s} \right]$$

$$\text{pressure coefficient} = C_p = \frac{\left[\frac{7M^2 \sin^2 \theta_s - 1}{6} \right]}{0.7\, M^2}$$

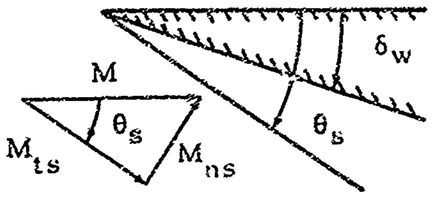
where

$(\ )_2$ = conditions behind the shock

Oblique shock detachment conditions are reached when no solution may be found to the above cubic relationships. Under these conditions the program uses the Newtonian + Prandtl-Meyer method for continued calculations.

Tangent-Wedge, Tangent-Cone, and Delta Wing Newtonian Empirical Method

The tangent-cone and the tangent-wedge Newtonian empirical methods used in this program are based on the empirical relationships derived below.



For wedge flow

$$\sin \theta_s = \frac{\sin \delta_w}{(1 - \epsilon) \cos (\theta_s - \delta_w)}$$

where

$$\epsilon = \frac{\rho}{\rho_2} = \frac{\gamma - 1}{\gamma + 1}\left[1 + \frac{2}{(\gamma - 1) M_{ns}^2}\right]$$

For cone flow (thin shock layer assumption)

$$\sin\theta_s \cong \frac{\sin \delta_c}{(1 - \frac{\epsilon}{2}) \cos (\theta_s - \delta_c)}$$

In the limit as $M \to \infty$, $\epsilon = \epsilon_{lim} = \frac{\gamma - 1}{\gamma + 1}$ and $\cos (\theta_s - \delta) = 1$

Therefore

wedge	cone
$\sin\theta_s = \frac{\gamma + 1}{2} \sin \delta_w$	$\sin \theta_s = \frac{2(\gamma + 1)}{\gamma + 3} \sin \delta_c$

These limiting expressions for θ may now be compared with the data of TR-1135 (Reference 25) at $\gamma = 7/5$ using the following similarity parameters. The exact equations contain three variables — θ_s, δ, and ϵ. Noting that for γ = constant, $\epsilon = fn(M_{ns})$ only, the preceding equations may be rewritten in the following form:

wedge

$$M_{ns} = \frac{M \sin \delta_w}{(1 - \epsilon) \cos (\theta_s - \delta_w)}$$

cone

$$M_{ns} = \frac{M \sin \delta_c}{(1 - \frac{\epsilon}{2}) \cos (\theta_s - \delta_c)}$$

The parameter $(\theta - \delta)$ is approximately constant and independent of M except near the shock detachment condition. The equations essentially contain only two variables, M_{ns} and $M \sin \delta$. These are used as coordinates to plot the data for wedge flow shown in Figure 4. A similar plot could be obtained for cone flow. From the figure it is seen that the data are nearly normalized with the use of these coordinates.

For rapid calculations we need relationships for M_{ns} as a function of $M \sin \delta$ that satisfy the following requirements:

1. The effect of shock detachment is neglected
2. At $M \sin \delta = 0$, $M_{ns} = 1$
3. The solution asymptotically approaches the $M = \infty$ line
4. Have the correct slope, $\frac{d\, M_{ns}}{d\, M \sin \delta}$ at $M \sin \delta = 0$

These conditions lead to equations of the following form

$$\text{wedge} \quad M_{ns} = K_w M' + e^{-\frac{K_w}{2} M'}$$

$$K_w = \frac{\gamma + 1}{2}$$

$$\text{cone} \quad M_{ns} = K_c M' + e^{-K_c M'}$$

where

$$M' = M \sin \delta$$

$$K_c = 2(\gamma + 1)/(\gamma + 3)$$

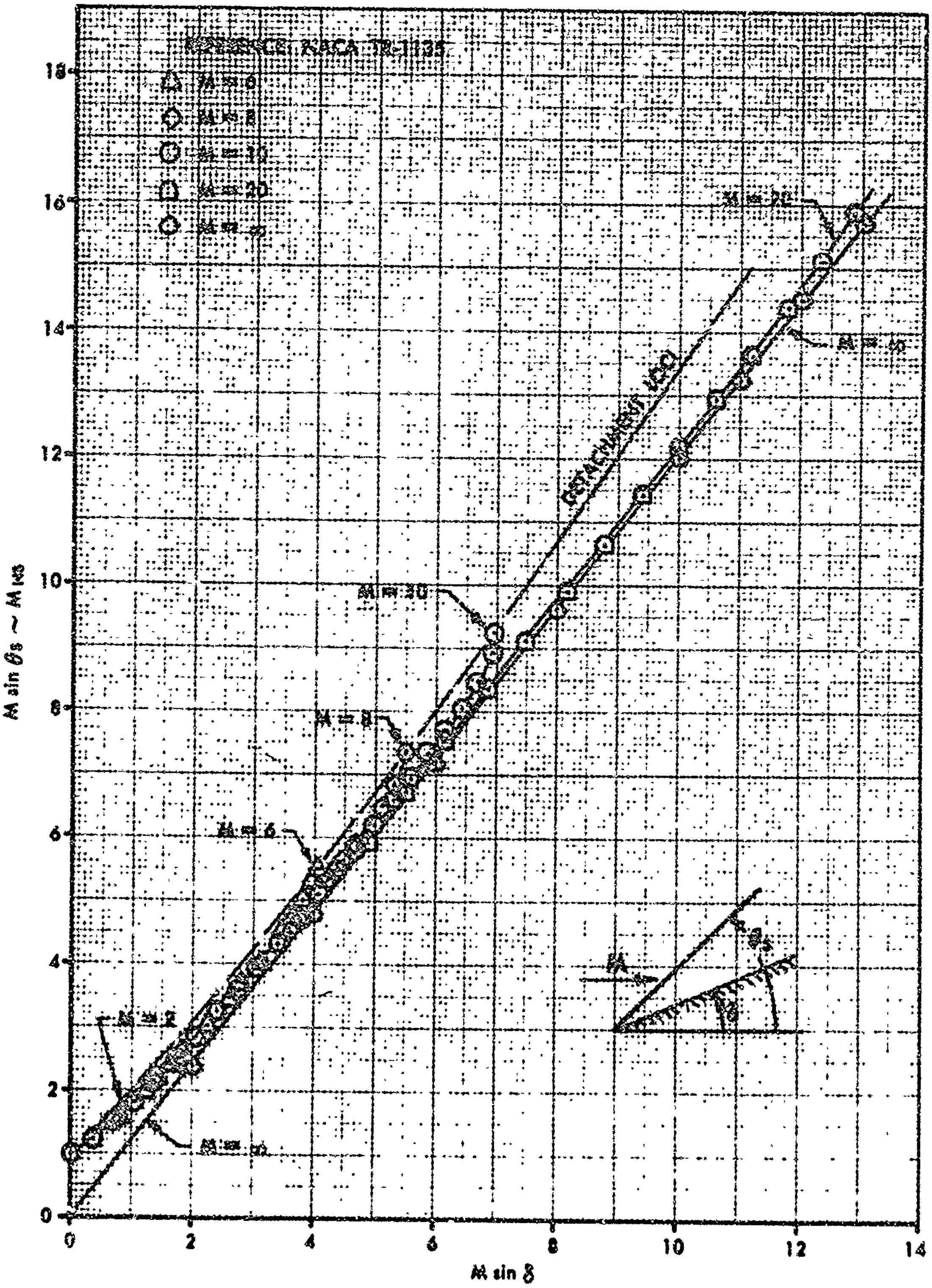


Figure 4. Wedge Flow Shock Angle

These expressions are compared with the data of TR-1135 in Figures 5 and 6. The cone data are also shown in Figure 7 with the same scales as in Figure 4.

The pressure coefficient may now be obtained by the following relationships for a wedge and cone respectively.

$$C_p = \left(\frac{4}{\gamma+1}\right)(M_{ns}^2 - 1) \;/\; M^2$$

$$C_p = 2 \sin^2\delta \left[1 - \frac{(\gamma-1)\, M_{ns}^2 + 2}{4\,(\gamma+1)\, M_{ns}^2}\right]^{-1}$$

Experimental results have shown the pressure on the centerline of a delta wing to be in agreement with two-dimensional theory at small values of the similarity parameter ($M' < 3.0$) and with conical flow theory at higher values. The previous expressions derived for wedge and cone flows have been combined to give these features. The resulting relationships are given below.

$$M_{ns} = K_C M' + e^{-(K_C - \frac{K_w}{2})\, M'}$$

For $\gamma = 7/5$

$$M_{ns} = 1.09\, M \sin\delta + e^{-0.49\, M \sin\delta}$$

The similarity parameter relationship for pressure is

$$M^2 C_p = \left(\frac{4}{\gamma+1}\right)\left(M_{ns}^2 - 1\right)$$

The shock angle and pressure coefficient calculated from the above equations are compared with the experimental results (Reference 28) in Figures 8 and 9, respectively.

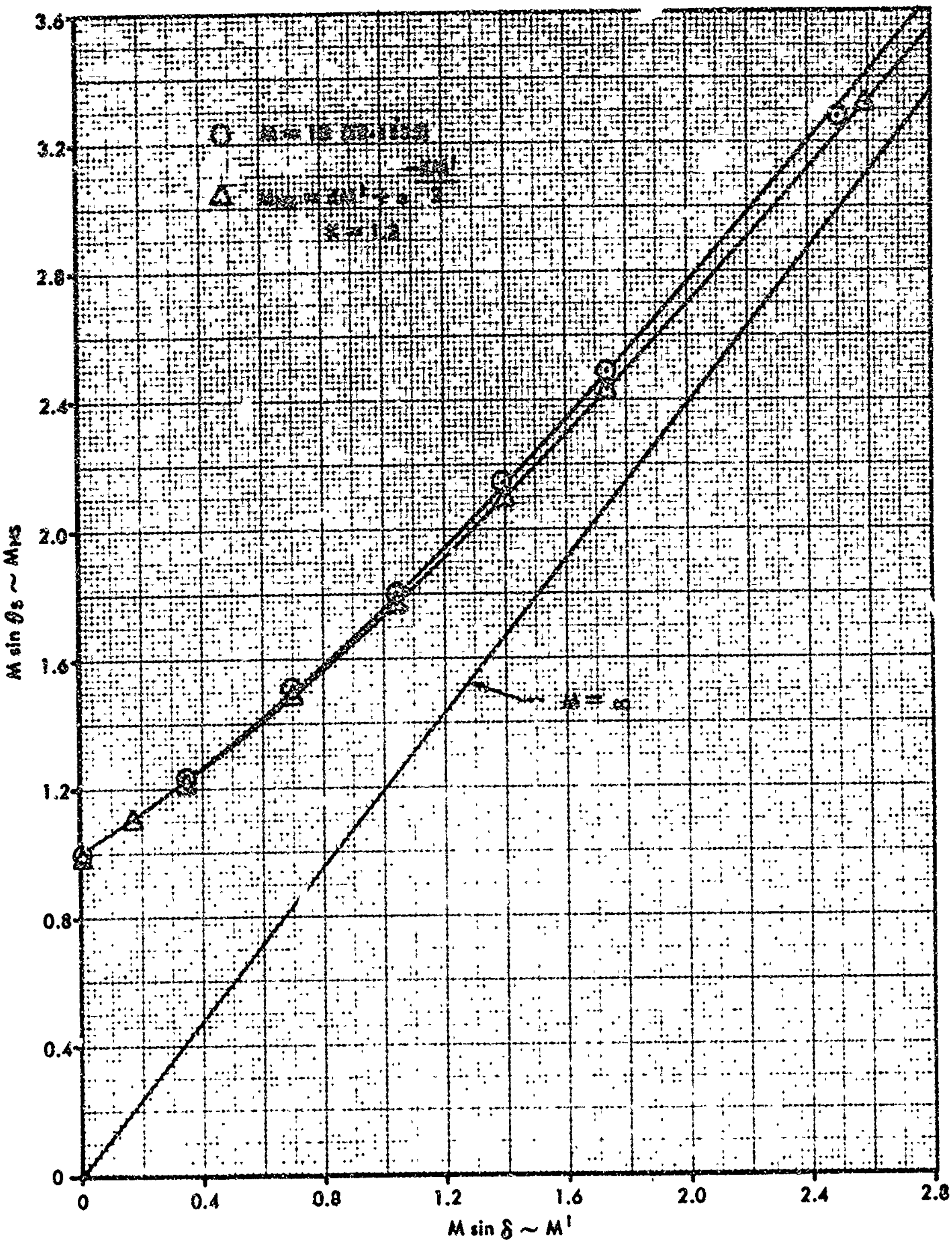


Figure 5. Wedge Flow Shock Angle Empirical Correlation

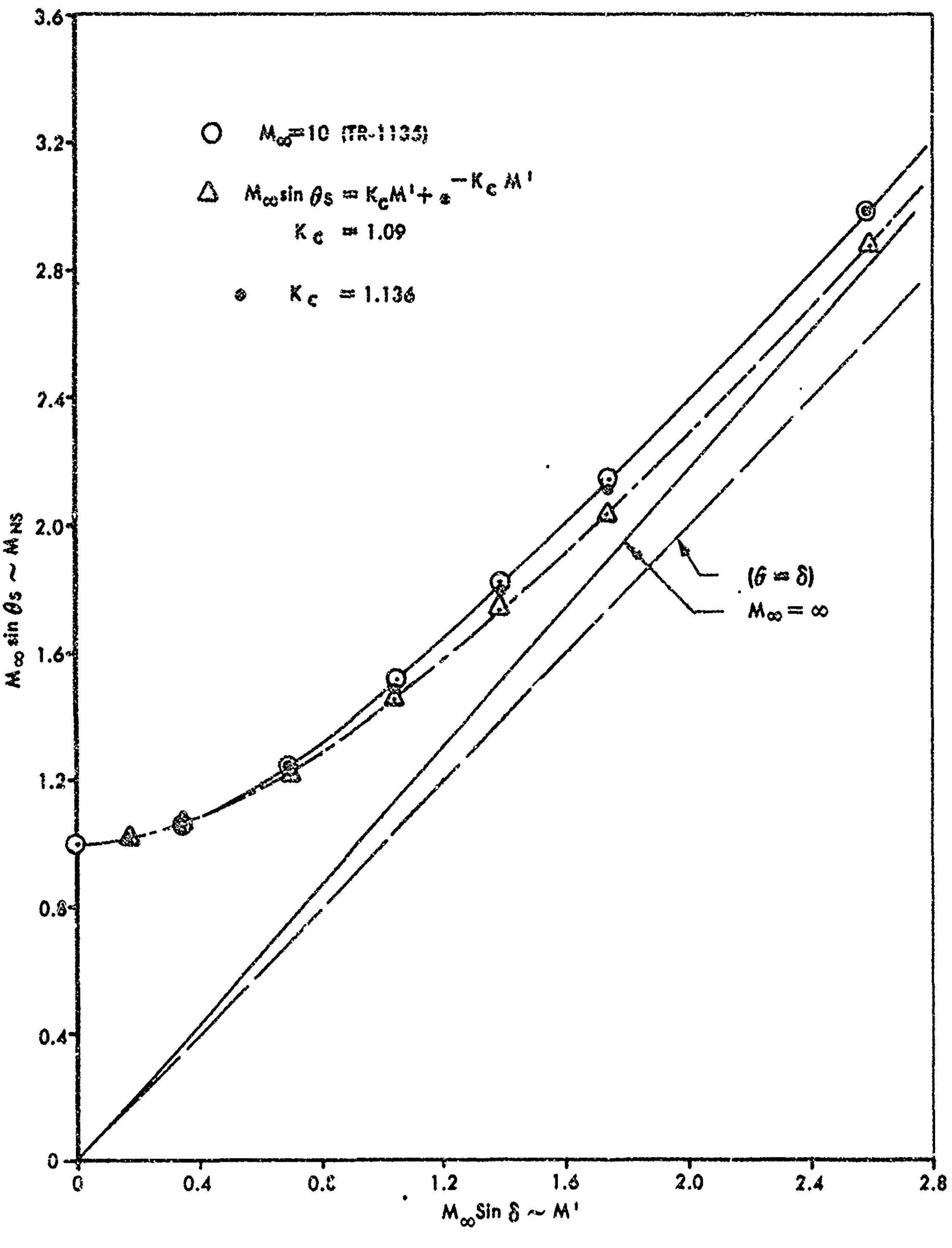


Figure 6. Conical Flow Shock Angle Empirical Correlation

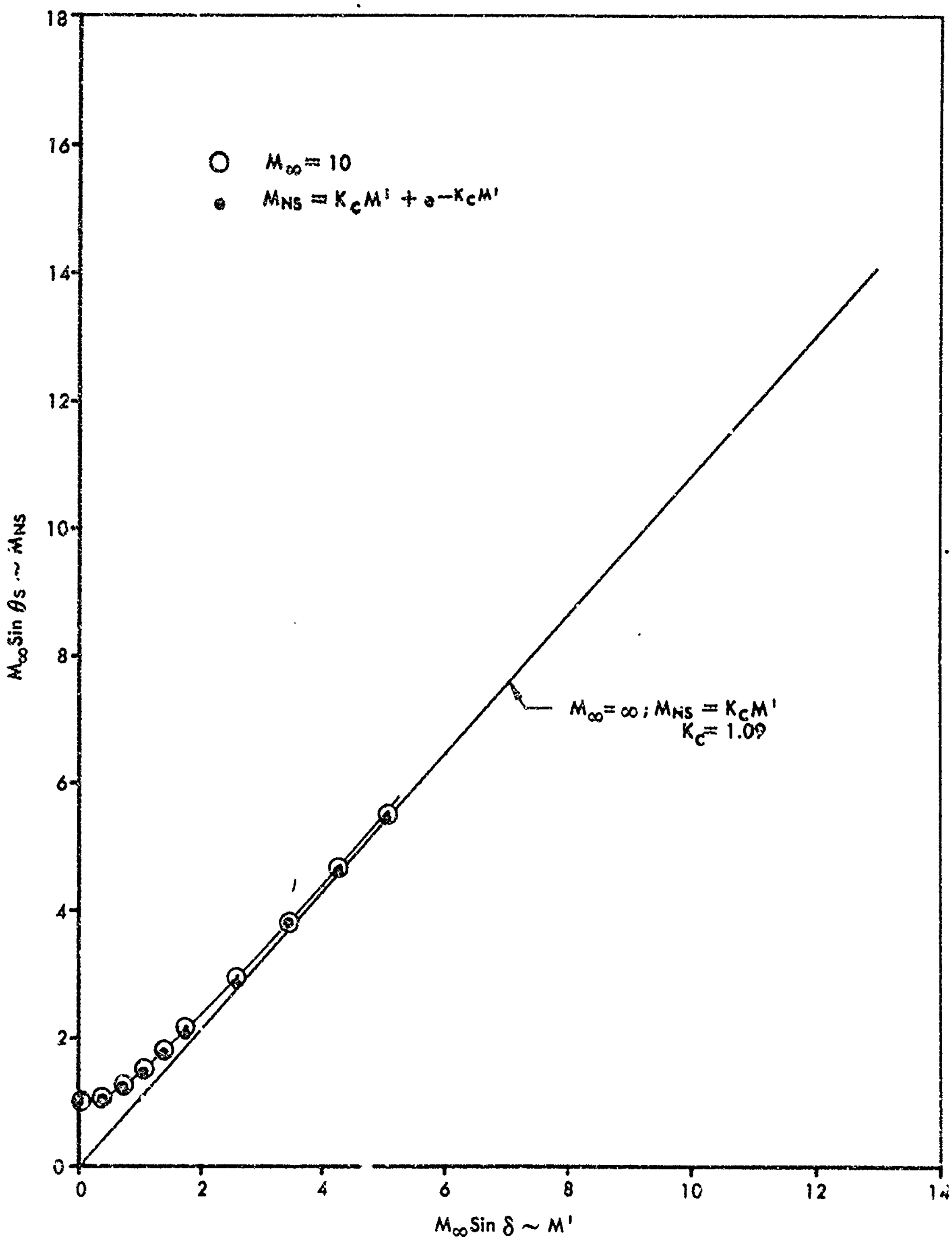


Figure 7. Conical Flow Shock Angle Empirical Correlation

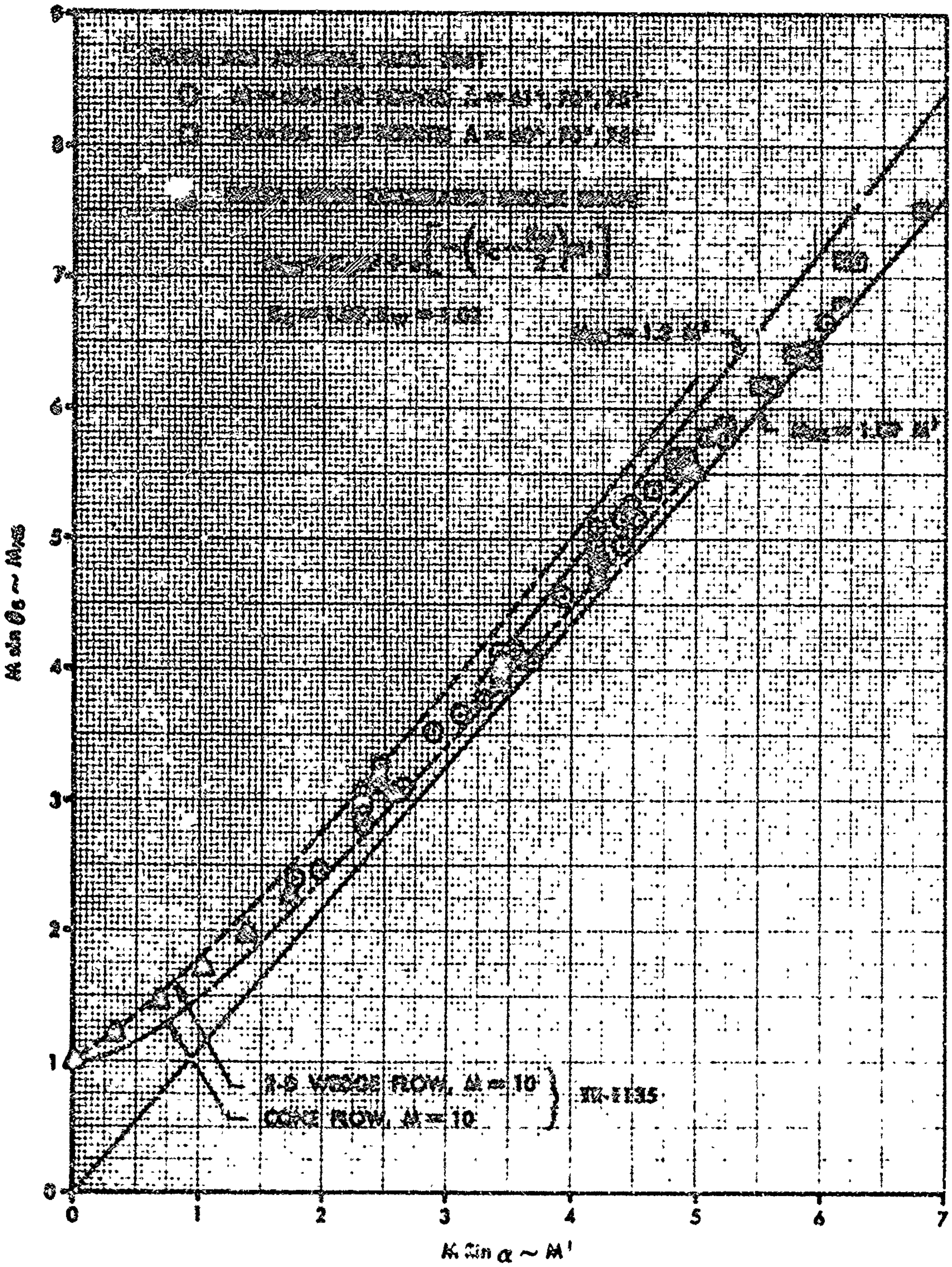


Figure 8. Delta Wing Centerline Shock Angle Correlation

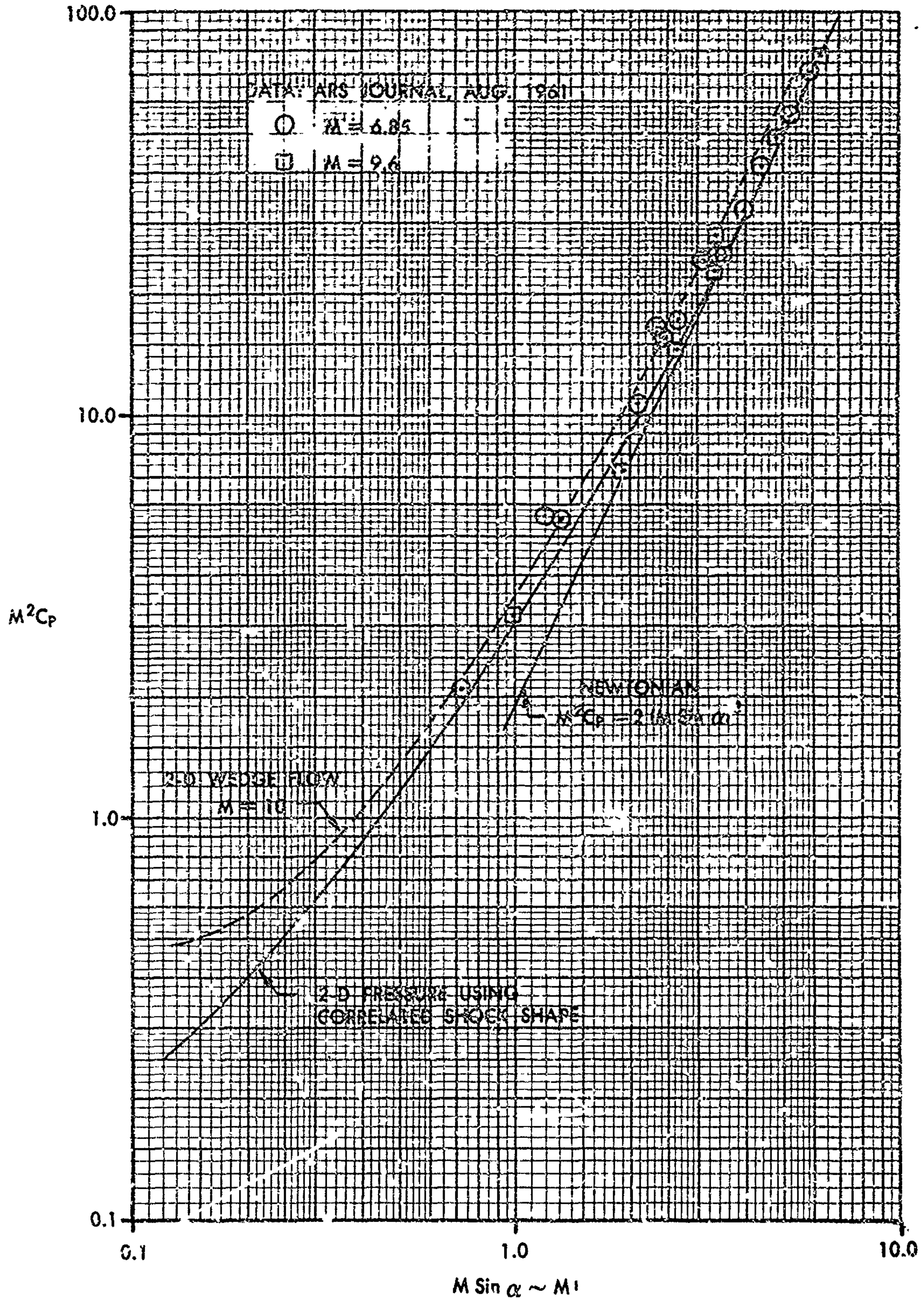


Figure 9. Delta Wing Centerline Pressure Coefficient Correlation

OSU Blunt Body Empirical Method

The OSU (Ohio State University) blunt body empirical equation describes the pressure distribution about cylinders in supersonic flow. The equation was presented in Reference 29 and was stated to match "all the data obtained on the cylinders in the present test series with a maximum deviation of 2.5 percent." The expression used is

$$\frac{p_l}{p_{t_\infty}} = 0.32 + 0.455 \cos\theta + 0.195 \cos 2\theta + 0.035 \cos 3\theta - 0.005 \cos 4\theta$$

where

θ = peripheral angle on a cylinder (= 0 at the stagnation point) = $90° - \delta$

p_l = surface pressure

p_{t_∞} = total pressure rise through normal shock

The pressure coefficient is calculated from the relationship

$$C_p = \left[\left(\frac{p_l}{p_{t_\infty}}\right)\left(\frac{p_{t_\infty}}{p_\infty}\right) - 1\right] \Big/ \left(\frac{\gamma}{2} M^2\right)$$

where

$\frac{p_{t_\infty}}{p_\infty}$ = $K \frac{\gamma}{2} M^2 + 1$

K = stagnation pressure coefficient = $C_{p_{stag}}$

p_∞ = freestream pressure

γ = ratio of specific heats = 1.4

Van Dyke Unified Method

This force calculation method is based on the unified supersonic-hypersonic small disturbance theory proposed by Van Dyke in Reference 30 as applied to basic hypersonic similarity results. The method is useful for thin profile shapes and as the name implies extends down to the supersonic speed region.

The similarity equations that form the basis of this method are derived by manipulating the oblique shock relations for hypersonic flow. The basic derivations are shown on pages 753 and 754 of Reference 31. The result obtained for a compression surface under the assumption of a small deflection angle and large Mach number is (hypersonic similarity equation).

$$C_p = \delta^2 \left[\frac{\gamma - 1}{2} + \sqrt{\left(\frac{\gamma + 1}{2}\right)^2 + \frac{4}{H^2}} \right]$$

where H is the hypersonic similarity parameter given by $M\delta$. The contribution by Van Dyke in Reference 30 suggests that this relationship will also be valid in the realm of supersonic linear theory if the hypersonic similarity parameter $M\delta$ is replaced by the unified supersonic-hypersonic parameter $\left(\sqrt{M^2 - 1}\right)\delta$. This latter parameter is used in the calculations for this force option in the arbitrary body program.

A similar method may also be obtained for a surface in expansion flow with no leading edge shock such as on the upper side of an airfoil. The resulting equation is

$$C_p = \delta^2 \frac{2}{\gamma H^2} \left[\left(1 - \frac{\gamma - 1}{2} H\right)^{\frac{2\gamma}{\gamma - 1}} - 1 \right]$$

where again H is taken to be $\left(\sqrt{M^2 - 1}\right)\delta$ in the unified theory approach.

Shock-Expansion Method

This force calculation method is based on classical shock-expansion theory (see Reference 31). In this method the surface elements are handled in a "strip-theory" manner. The characteristics of the first element of each longitudinal strip of elements may be calculated by oblique shock theory, by conical flow theory, or by a Prandtl-Meyer expansion. Downstream of this initial element the forces are calculated by a Prandtl-Meyer expansion.

By a proper selection of the element orientation the method may be used for both wing-like shapes and for more complex body shapes. In this latter case the method operates in a hypersonic shock-expansion theory mode.

Free Molecular Flow Method

At very high altitudes conventional continuum flow theories fail and one must begin to consider the general macroscopic mass, force, and energy transfer problem at the body surface. This condition occurs when the air is sufficiently rarefied so that the mean free path of the molecules is much greater than a characteristic body dimension. This condition is known as free molecular flow and the method of analysis selected for this program is described in Reference 32. This method was also used in Reference 19. The equations used were taken from these references and are presented below.

Pressure coefficient

$$C_p = \frac{1}{S^2}\left\{\left[\frac{2-f_n}{\sqrt{\pi}} S \sin\delta + \frac{f_n}{2}\sqrt{\frac{T_b}{T_\infty}}\right] e^{-(S\sin\delta)^2} + \left[(2-f_n)\left(S^2\sin^2\delta + \frac{1}{2}\right) + \frac{f_n}{2}\sqrt{\pi}\sqrt{\frac{T_b}{T_\infty}}\, S\sin\delta\right]\left[1 + \operatorname{erf}(S\sin\delta)\right]\right\}$$

Shear force coefficient

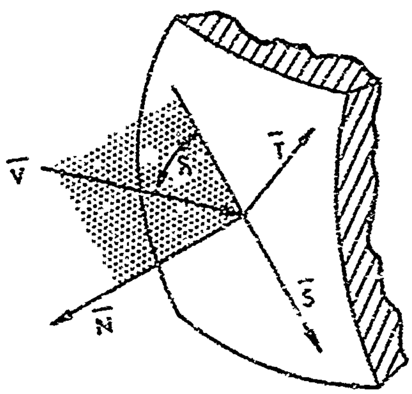
$$C_f = \frac{(\cos\delta) f_t}{\sqrt{\pi} S}\left\{e^{-(S\sin\delta)^2} + \sqrt{\pi} S \sin\delta\left[1 + \operatorname{erf}(S\sin\delta)\right]\right\}$$

where

S	=	speed ratio = $\sqrt{\gamma/2}\ M_\infty$
f_n	=	normal momentum accommodation coefficient (= 0.0 for Newtonian and = 1.0 for completely diffuse reflection)
δ	=	impact angle
T_b	=	body temperature, °K
T_∞	=	free-stream temperature, °K
erf	=	error function $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-x^2}\,dx$
f_t	=	tangential momentum accommodation coefficient (= 0. for Newtonian flow and 1.0 for completely diffuse reflection)

The pressure force acts perpendicular to the surface and this direction is readily obtained since the element normal has already been determined in the geometry subroutines. The shear force acts in the direction of the tangential velocity component on the surface and this direction is determined by taking successive vector products as follows.

The procedure is illustrated in the accompaning sketch where the incident velocity vector is defined as



$$\overline{V} = V_X \overline{i} + V_Y \overline{j} + V_X \overline{k}$$

and the surface normal as

$$\overline{N} = N_X \overline{i} + N_Y \overline{j} \, N_Z \overline{k}$$

First, a surface tangent vector $(\overline{T})$ is defined by the cross product of the normal and velocity vectors:

$$\overline{T} = T_X \overline{i} + T_Y \overline{j} - T_Z \overline{k}$$

where

$$T_X = N_Y V_Z - N_Z V_Y$$

$$T_Y = N_Z V_X - N_X V_Z$$

$$T_Z = N_X V_Y - N_Y V_X$$

Then the direction of the shear force $(\overline{S})$ is given by the cross product of the surface tangent and normal vectors;

$$\overline{S} = S_X \overline{i} + S_Y \overline{j} + S_Z \overline{k}$$

where

$$S_X = T_Y N_Z - T_Z N_Y$$

$$S_Y = T_Z N_X - T_X N_Z$$

$$S_Z = T_X N_Y - T_Y N_X$$

The final components of the shear force in the vehicle axis system are given by

$$SHEAR_X = (SHEAR)\left(S_X\right) / STOTAL$$

$$SHEAR_Y = (SHEAR)\left(S_Y\right) / STOTAL$$

$$SHEAR_Z = (SHEAR)\left(S_Z\right) / STOTAL$$

where

SHEAR is the shear force as calculated by the free molecular flow equations.

$$STOTAL = \left(S_X^2 + S_Y^2 + S_Z^2\right)^{1/2}$$

In using the free molecular flow method the above analysis must be carried out over the entire surface of the shape including the base, shadow regions, etc. When the free molecular flow method is selected, it is used for both impact and shadow region.

This method of determining the shear direction is also used for the continuum viscous forces discussed in the next section. The plane formed by the velocity vector and the surface normal is referred to as the velocity plane (shaded region in the sketch), since both the incident and surface velocity are in this plane. This definition is correct for two-dimensional flow, however, it is only an approximation to the shear direction in the general arbitrary-body case.

Hankey Flat-Surface Emprical Method

This method uses an empirical correlation for lower surface pressures on blunted flat plates. The method, derived in Reference 14, approximates tangent-wedge at low impact angles and approaches Newtonian at high impact angles. The pressure coefficient is given by

$$C_p = 1.95 \sin^2 \delta + 0.21 \cos \delta \sin \delta$$

Dahlem-Buck Empirical Method

This is an impact method that has been derived such that tangent-cone and Newtonian results are approximated, respectively, at low and high values of the impact angle. The empirical relationships presented in Reference 33 are

for $\delta < 22.5°$ $$C_p = \frac{1 + (\sin 4\delta)^{3/4}}{(4 \cos \delta \cos 2\delta)^{3/4}} (\sin \delta)^{5/4}$$

for $\delta \geq 22.5°$ $$C_p = 2.0 \sin^2 \delta$$

Blast Wave Pressure Increments

This method uses conventional blast-wave parameters to calculate the overpressure due to bluntness effects. Force contributions determined by this procedure must be added to the regular inviscid pressure forces (tangent-wedge, tangent-cone, Newtonian, etc.) calculated over the same vehicle geometry. The specific blast wave solutions used in the Program were derived by Lukasiewicz in Reference 34:

$$\frac{P}{P_\infty} = A\, M_\infty^2 \left\{ \frac{(C_D)^{\frac{1}{1+j}}}{(X_o - X)/d} \right\}^{\frac{2+j}{3}} + B$$

where

C_D is the nose drag coefficient

d is the nose diameter or thickness

X_o is a coordinate reference point

and the coefficients A, B are

Flow	j	A	B
Two-Dimensional	0	0.121	0.56
Axisymmetric	1	0.067	0.44

Modified Tangent-Cone Method

This method. originally developed for use on cones with elliptical cross-sections, modifies the tangent-cone result by an increment representing the deviation from an average pressure divided by an average Mach number. More specifically, the following equations are used (after Jacobs, Reference 35):

$$C_p = C_{p_{tc}} - \frac{C_{p_{tc}} - C_{p_{avg}}}{M_{avg}}$$

where C_p is the surface pressure coefficient

$C_{p_{tc}}$ is the conventional tangent-cone pressure coefficient

$C_{p_{avg}}$ is the average pressure coefficient

$= \sum C_{p_{t}} A / \sum A$, A is element area

M_{avg} is the average Mach number, defined for an equivalent cone having pressure coefficient $C_{p_{avg}}$.

High Mach Base Pressures

For a body in high speed flow it might be expected that any base regions would experience total vacuum. That is,

$$C_p = - \frac{1}{\frac{\gamma}{2} M_\infty^2}$$

However, the viscosity of real gases causes some pressure to be felt in base region and experimental data have shown this to be roughly 70% vacuum for air. Therefore, the expression

$$C_p = - \frac{1}{M_\infty^2}$$

has been included in the program.

Viscous Force Calculation Methods

The most difficult part in the analysis of an arbitrary shape is the calculation of viscous forces. A detailed knowledge of the local properties and the flow history along surface streamlines is required. This combined with the natural complexity of the boundary-layer equations necessitates considerable simplification of the problem before solutions can be obtained. An engineering approach has been selected that retains the essential characteristics of the hypersonic boundary-layer problem. No attempt is made to calculate the detailed skin friction distribution on the exact arbitrary shape, but rather, the vehicle is represented by a number of flat surfaces on each of which the shear force is determined.

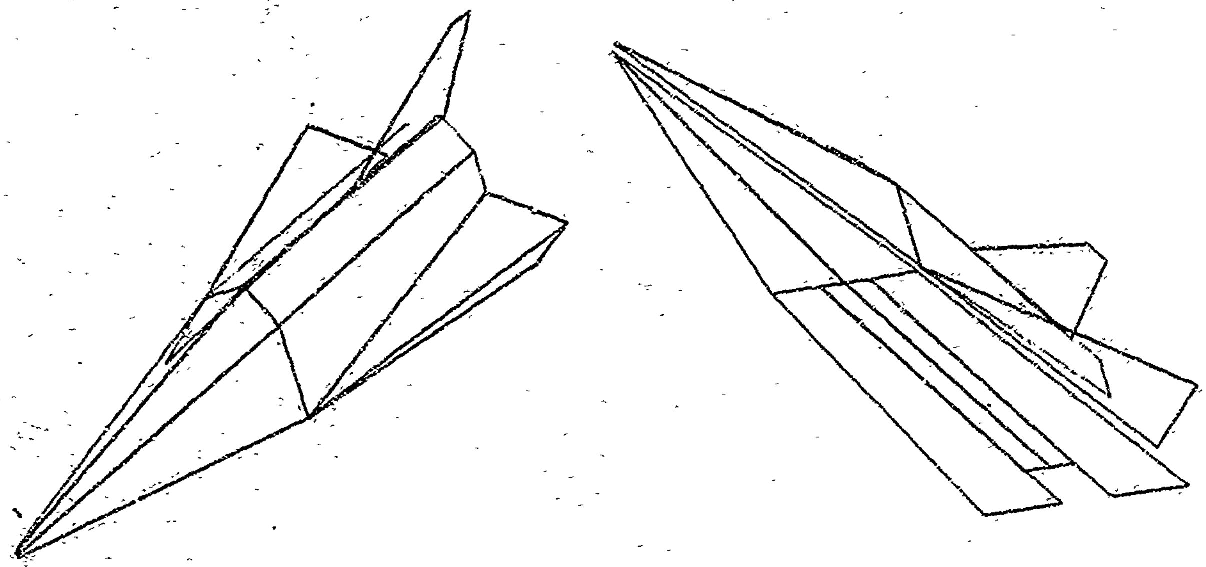
The surface streamlines are assumed in the velocity plane and the flow history is approximated by the inclusion of an initial surface. The shear force is determined for both laminar and turbulent flow and may be summed over the vehicle for either type.

Reference temperature and reference enthalpy methods are available for both laminar and turbulent flows and, in addition, the Spalding-Chi method with either temperature or enthalpy ratios may be selected for turbulent calculations. The surface temperature may be either input or the radiation equilibrium value determined. The effect of planform shape, leading edge viscous-interaction, and the viscous forces on blunt bodies are also considered.

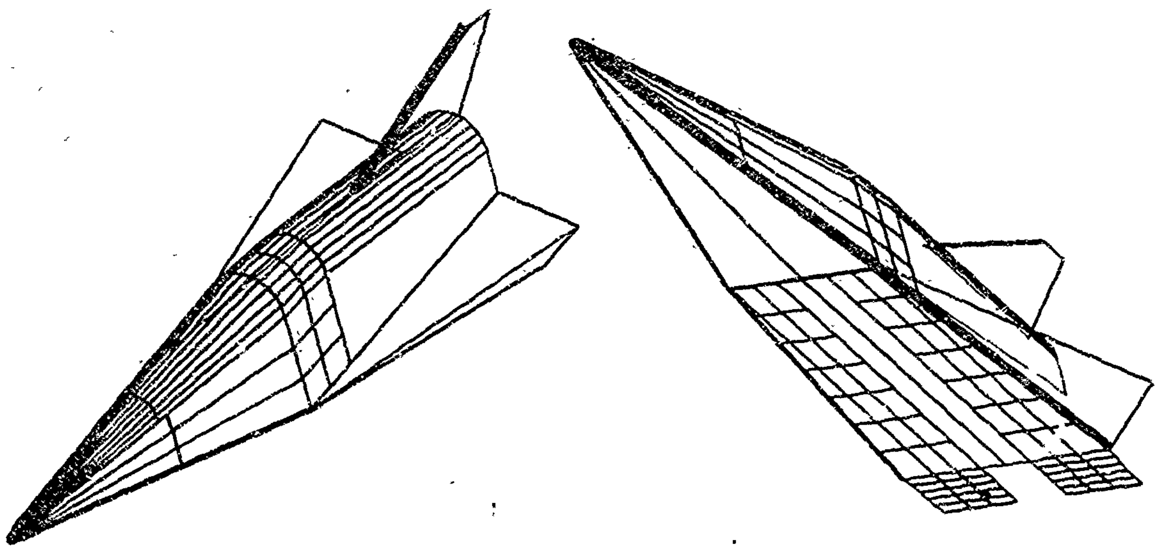
Skin Friction Geometry Model

For the skin friction calculations a geometrically complex vehicle is divided up into a number of plane surfaces in a manner which adequately approximates the true shape. Leading-edge surfaces and local curvatures are omitted. Regions of relatively large curvatures can be represented by using a greater number of plane surfaces. The degree to which this is done will depend upon the complexity of the actual shape and experience of the designer. The geometry data for the skin-friction geometry model is prepared in the same way as the surface element data used for the inviscid pressure calculations and retain their relative location to each other and to the flight path. This skin friction modeling technique is best described by viewing, for example, a typical high L/D vehicle shown in Figure 10. The upper half presents the skin-friction representation of the vehicle which is to be contrasted with the detailed inviscid geometry given in the lower half of the figure. As used in the in the Hypersonic Arbitrary-Body Program the skin friction surfaces are referred to as an approximate representation of the vehicle geometry while, in fact, it has been observed they are as complete as generally used throughout the industry for the inviscid calculations.

From the input element data, the surface normal, area, and area centroid coordinates are calculated. In addition, maximum chord length, taper ratio, and true area are input for each surface. The latter may be different from the calculated area since curvatures have been neglected. The initial surface, specified by its maximum chord length and taper ratio, is assumed to be in the plane of the skin-friction surface and, therefore, the flow history is only approximated. The element planform effect on the average skin friction is



a) Representation for Viscous Calculations



b) Representation for Inviscid Calculations

Figure 10. Geometry Modeling for a Typical High L/D Vehicle

included, however, and is discussed separately for laminar and turbulent flows in later sections. The shear force on each surface is assumed to act through its centroid in a direction on the surface parallel to plane containing the surface normal and the free-stream velocity vector, as described in the section on free-molecular flow.

Local Flow Conditions

The required local properties (pressure, temperature, density, and velocity) are obtained assuming a calorically perfect gas. The pressure on each skin-friction surface may be determined from a choice of several of the inviscid pressure methods — tangent-wedge, tangent-cone, Newtonian+Prandtl-Meyer, and Prandtl-Meyer expansion. At the present time, a continuous strip shock-expansion calculation is not available within the Arbitrary-Body Program and, in this respect, each surface is treated independently of the others.

The skin-friction surfaces and local properties, thus, have been defined in a way that reduces the problem of calculating the viscous forces on a complex shape to one of solving for the skin friction on a number of constant-property flat plates.

Incompressible Flow

The basic philosophy behind both the Spalding-Chi and the reference condition methods is the same. Namely, that the suitably transformed skin-friction coefficient is given by the constant-property or incompressible formulas based on a Reynolds number also suitably transformed. To emphasis the point, this may be stated another way: The compressible skin-friction is given by the incompressible form with appropriate correction factors to account for compressibility effects. That is,

$$C_{f_\delta} = C_{f_i}/F_c$$

$$C_{f_i} = f(Rx_i)\ ,\quad Rx_i = F_{Rx} \bullet Rx$$

where

C_f = skin friction coefficient

Rx = Reynolds number

$(\ )_i$ = indicates incompressible

$(\ )_\delta$ = indicates compressible

The incompressible formulas used in the Hypersonic Arbitrary-Body Program are given in Table I and the compressibility factors, F_c and F_{Rx} are discussed below.

Flow	Skin Friction Coefficient, $f(Rx_i)$		Source
	Local	Average	
Laminar	$0.664/\sqrt{Rx_i}$	$1.328/\sqrt{Rx_i}$	Blasius
Turbulent ($Rx_i > R_{Min}$)	$\dfrac{0.088\,(\log Rx_i - 2.3686)}{[\log Rx_i - 1.5]^3}$	$\dfrac{0.088}{[\log Rx_i - 1.5]^2}$	Sivells & Payne (Ref. 36)
R_{Min}	2540	6570	

Table I. Incompressible Skin-Friction Coefficient Formulas

The Sivells and Payne formulas have singularities occurring at low Reynolds numbers. However, both occur below the point at which the turbulent values cross the respective Blasius laminar curves. Thus, the turbulent incompressible skin-friction coefficients for Reynolds numbers equal to or less than R_{Min} are given by the corresponding laminar values.

Compressible Flow

Reference Temperature and Reference Enthalpy Method

$$F_c = \rho_\delta/\rho^*$$

$$F_{Rx} = (\mu_\delta/\mu^*)\,\frac{1}{F_c}$$

where ρ is the density, μ the viscosity, and the superscript "*" means evaluated at the reference temperature, T^*, or reference enthalpy, H^*;

$$\frac{T^*}{T_\delta} = (A1)\,\frac{T_W}{T_\delta} + (A2)\,\frac{T_{AW}}{T_\delta} + (1 - A1 - A2)$$

$$\frac{H^*}{H_\delta} = (A1)\,\frac{H_W}{H_\delta} + (A2)\,\frac{H_{AW}}{H_\delta} + (1 - A1 - A2)$$

The value of the coefficients used are due to Monaghan (Reference 37) for Prandtl number equal to 0.71;

$$A1 = 0.5825$$

$$A2 = 0.1875$$

The subscript "W" indicates the wall value and subscript "AW" refers to adiabatic wall conditions given by

$$\frac{T_{AW}}{T_\delta} = \frac{H_{AW}}{H_\delta} = 1 + \left(\frac{\gamma - 1}{2}\right) r \ M_\delta^{2}$$

where

γ = ratio of specific heats (= 1.4)

M = Mach number

r = recovery factor = $(P_r)^{1/n}$

n = 2 for laminar flow

n = 3 for turbulent flow

P_r = Prandtl number (= 0.71)

Spalding-Chi Method (Reference 38):

$$F_c = A \Big/ \left\{ \mathrm{ARSIN}\left(\frac{A-B}{C}\right) + \mathrm{ARSIN}\left(\frac{A+B}{C}\right) \right\}^2$$

where

$$A = \frac{H_{AW}}{H_\delta} - 1$$

$$B = \frac{H_W}{H_\delta} - 1$$

$$C = \left[(A+B)^2 + 4A\right]^{1/2}$$

$$F_{Rx} = \left(\frac{H_{AW}}{H_\delta}\right)^{q} \Big/ \left[F_c\left(\frac{H_W}{H_\delta}\right)^{p+q}\right], \quad q = 0.772, \quad p = 0.702$$

Surface Equilibrium Temperature

In the Arbitrary-Body Program the surface equilibrium temperature is defined as the temperature satisfying the steady-state heat balance between the boundary-layer convection to the surface and the surface radiation to space.

convective heating: $QC(T_c) = C_h \ (H_{AW} - H_W)$

radiation heating: $QR(T_r) = R_K \ T_R^{\ 4}$

where C_h is the heat transfer coefficient

and $R_K = \epsilon\sigma$, ϵ = emissivity (= 0.8)

σ = Stefan-Boltzman constant

The surface equilibrium temperature is defined when $QC(T_C) = QR(T_R)$ for $T_C = T_R$. The solution is obtained by a simple linear intercept technique illustrated in the sketch and explained briefly as follows.

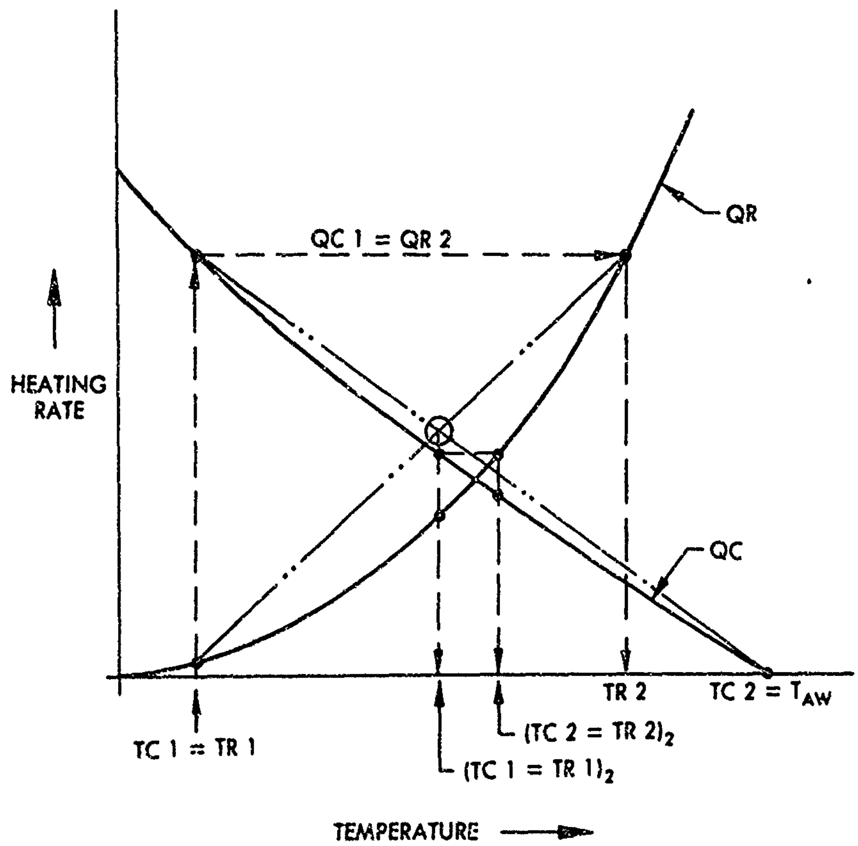
Linear relations are assumed for both heating rates

$$QC = AC + (BC)T$$

$$QR = AR + (BR)T$$

The four coefficients are initialized as follows.

1. Assume $TC1 = TR1 = 100\ ^{o}R$
2. Calculate QC1 and QR1
3. Let QR2 = QC1 and calculate
$$TR2 = \left(\frac{QR2}{R_K}\right)^{1/4}$$
4. If $TR2 > TC2 = T_{AW}$, then set $TR2 = TC2$ and calculate new QR2



The coefficients may now be readily determined and the result of the linear solution of the heat balance equation is simply

$$T = (AC - AR)/(BR - BC)$$

The convective and radiation heating rates are then calculated at this temperature and checked for convergence:

$$\left|1 - QC1/QR1\right| \leq EPST, \text{ where } EPST = 5.0E\text{-}4$$

If the criteria is not satisfied the cycle is repeated with TC1 = TR1 = T, QR2 = QC1, and TC2 = TR2. The present technique, while lacking sophistication, is accurate and quite rapid. Normally, two or three cycles are required for ideal gas solutions and one additional cycle for real gas cases.

Real Gas Effects

It is felt that some comments are in order with regard to the overall procedure. Specifically, what is the correctness or justification in using real gas reference enthalpy viscous solutions when the local inviscid flow has been determined only for a calorically perfect or ideal gas? To answer this question, an extensive comparison of laminar boundary-layer methods was undertaken in support of an earlier study and the details are reported in Reference 39. Briefly, the skin friction was determined for the flight conditions of the matrix given in Table II, corresponding to the surface equilibrium temperatures (emissivity = 0.8) at the one-foot station of a flat plate.

Altitude (1000 Ft)	Velocity (1000 fps)					
	8	12	16	20	24	28
100	x	x	x	x	-	-
150	-	x	x	x	x	-
200	-	-	x	x	x	x
250	-	-	x	x	x	x

Table II. Flight Matrix for Skin Friction Calculations

Angle-of-attack variation from 0° to 40° in 10° increments and five boundary-layer calculations were made at each condition. The latter correspond to the combination of three boundary-layer solutions and two shock wave solutions for local properties as shown in Table III.

Boundary Layer Solution	Local Properties	
	Real	Ideal
Exact	1	-
Reference Enthalpy	2	3
Reference Temperature	4	5

Table III. Boundary Layer Calculations

Also, additional calculations were made at the flight condition of 20,000 fps, 200,000 feet altitude, and wall temperature equal to 2000°R.

Methods 1 , 2 , and 5 are self-consistent with respect to the assumptions made and are regarded as normal calculation modes. Methods 3 and 4 are inconsistent in the assumptions made between the inviscid and viscous solutions and are termed mixed calculation modes. The free-stream properties were specified by the 1962 U.S. Standard Atmosphere and Sutherland's viscosity formula. The oblique shock-wave solutions are accurate to 5-significant digits in the inverse density ratio. For the real gas solution, the thermodynamic properties for equilibrium dissociating and ionizing air were obtained by the method in Reference 40. The assumed ideal gas is calorically perfect with ratio of specific heats equal to 1.40.

The real gas variation for the density-viscosity product in the viscous solutions was obtained as a function of enthalpy and pressure using the polynominal equations given in Reference 41. This product is based on the most recent thermodynamic data of Hilsennath (Reference 42) and the viscosity calculations of Hansen (Reference 43). The Prandtl number was assumed equal to 0.71 for all the methods.

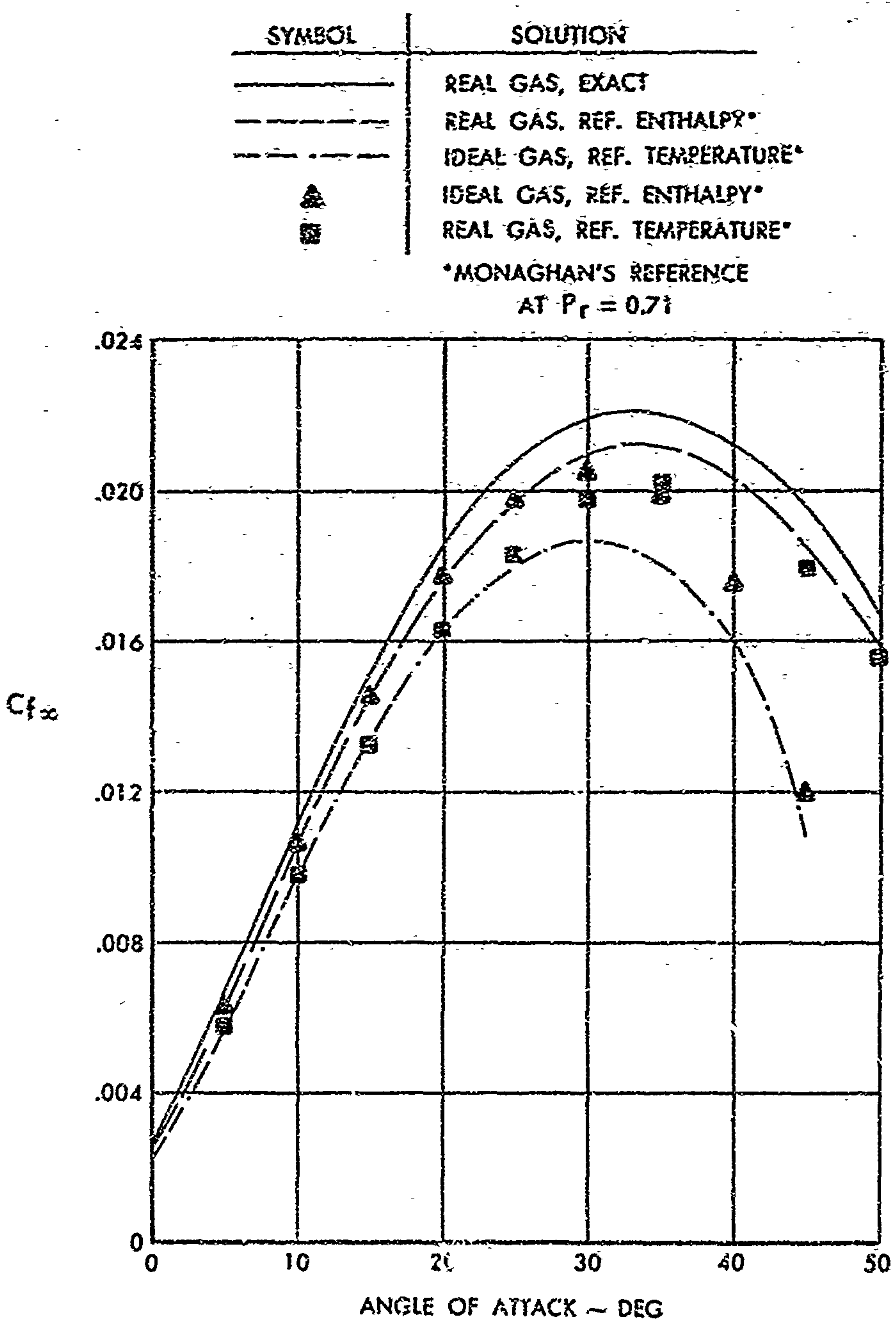


Figure 11. Laminar Skin-Friction Coefficient Comparison
(Altitude = 200,000 Ft., Velocity = 20,000 fps., T_W = 2000°R)

Typical results of the comparison are shown in Figure 11. The exact solutions were obtained using the Douglas General Laminar Compressible Boundary-Layer Program as described in Reference 41. The reference method calculations shown are based on the coefficient values of Monaghan. These were selected since the skin friction calculated consistently gave the best agreement with the exact results. Comparison of the three formulations considered — Monaghan (Reference 37), Michel (Reference 44) and Eckert (Reference 45) are shown in Table IV for the same flight conditions as Figure 11. Major conclusions of the comparison are:

1. With the exception of possibly zero angle-of-attack the reference temperature method, using existing values for the coefficients A1 and A2, is inadequate for predicting skin friction for the complete range of hypersonic flight conditions considered.

2. The real gas, reference enthalpy method using Monaghan's formulation adequately predicts the laminar skin friction over the complete flight range considered. The results, however, are consistently about 3 to 5 percent lower than the exact calculations.

3. The mixed calculation mode, ideal gas inviscid — real gas reference enthalpy is in substantial agreement with the real gas reference enthalpy calculation up to 30° angle-of-attack.

Reference Enthalpy Due to	Angle of Attack in Degrees									
	0	5	10	15	20	25	30	35	45	50
Monaghan	0.247	0.623	1.056	1.445	1.753	1.969	2.096	2.121	1.853	1.590
Michel	0.247	0.628	1.062	1.447	1.747	1.953	2.067	2.075	1.788	1.529
Eckert	0.243	0.613	1.038	1.418	1.717	1.926	2.042	2.058	1.788	1.534

Table IV Comparison of Reference Methods. Values of $C_f \times 10^2$.
(Altitude = 200,000 Ft., Velocity = 20,000 fps, T_W = 2000°R)

On the basis of the results of this study, the mixed-mode ideal gas inviscid-real gas reference enthalpy calculation has been included in the Hypersonic Arbitrary-Body Program. The real gas fluid properties of air are determined by the procedures described in detail in Reference 41. Three different formulas are used to specify the viscosity. At very low temperatures such as might be experienced in a high speed wind tunnel the viscosity is found from the

Bromley-Wilke results (Reference 46). In the Arbitrary-Body Program these are approximated by the following linear relationship;

For $T \leq 225^oR$

$$\mu = 0.80383436\ T \times 10^{-9}\ \frac{lb\ sec}{ft^2}$$

At higher temperatures and for an ideal gas the Sutherland viscosity formula is used (Reference 25):

For $T > 225^oR$

$$\mu = 2.270\ \frac{T^{3/2}}{T + 198.6} \times 10^{-8}\ \frac{lb\ sec}{ft^2}$$

For real gas and temperatures greater than about 6000^oR Hansen's viscosity values are used (Reference 43).

Viscous - Inviscid Interaction

Under conditions of low Reynolds number and high Mach number, the mutual interaction of the boundary layer and the inviscid flow field can have a large effect on both the laminar skin friction and surface pressure. Boundary-layer displacement effects in hypersonic flow over flat plates have been studied at length (e.g., Reference 47) and the present approach is limited to consideration of these methods. Basically, a pressure is induced from the relatively large outward streamline deflection caused by the thick hypersonic boundary layer. The classical approach is to consider an effective body, made up of the actual body plus the boundary-layer displacement thickness, in an iterative solution with the inviscid flow. This in itself is an approximation and, in addition, the simplifying assumptions of hypersonic viscous similarity are usually employed. This procedure has been adopted for use in the Arbitrary-Body Program and a brief background and development of the final equations follow.

Bertram and Blackstock (Reference 48) presented some simple procedures for estimating the boundary layer induced effects on pressure and skin friction. These involved the use of hypersonic-similarity-boundary-layer theory solutions in an iterative technique with the hypersonic small-disturbance tangent-wedge pressure equation. The analysis showed good correlation with experimental data for surfaces at nearly zero degrees incidence to the free-stream. White (Reference 49) extended the theory of Bertram and Blackstock to include the effect of angle of attack and presented a direct method for solving the problem without requiring iterations. White used hypersonic small disturbance expressions for both compression and expansion flows and introduced a new interaction parameter to correlate the wall temperature effect. Recently, Bertram (Reference 50) has presented more elaborate solutions for the problem employing the techniques of White. Implicit to all these solutions is the assumption of a calorically perfect gas and a Prandtl number of unity.

White's solution has been used in the present analysis because of the relative simplicity in its application. His numerical results showed the local pressure to be nearly a linear function of the iteraction parameter, λ;

$$P = P_o (1 + B)$$

where

$$B = m \frac{\lambda}{P_o} ,$$

and

$$\lambda = \frac{G M_\infty^3}{\sqrt{1+2j}} \left(\frac{C}{Rx}\right)^{1/2}$$

The quantity G is a simple function of wall temperature and specific heat, C is the Chapman-Rubesin viscosity coefficient, and j is the Manger transformation parameter: two-dimensional flow, $j = 0$; axially-symmetric flow, $j = 1$.

In the above equations, P is the local pressure to free-stream pressure ratio, and the subscript "o" refers to the inviscid value obtained from the hypersonic small-disturbance relations.

Bertram's (Reference 48) correlation for local skin friction coefficient is

$$C_f = 0.664\ K_1 \left(\frac{PC}{Rx}\right)^{1/2}$$

where K_1 is a pressure gradient and wall temperature correction factor. The shear on the surface is

$$\tau_W = \int q_\delta C_f \, dA$$

In the present analysis, the approach taken is to determine the effect or factor due to viscous-interaction using White's method and then to modify the previous result without interaction accordingly. This viscous-interaction factor, K_{VI}, is obtained by carrying out the integration of the preceding equation and is defined as follows;

$$K_{VI} = \frac{(\tau_W)_{VI}}{\tau_W} = \sqrt{1 + B_{c_r}} + B_{c_r} \log_e \left| \frac{\sqrt{1 + B_{c_r}} + 1}{\sqrt{B_{c_r}}} \right|$$

where B_{c_r} is based on the root-chord and K_1 has been assumed equal to one. This expression is for a plate with taper ratio one, but the integration could have been done for an arbitrary value (e.g., Reference 51). In the present application the planform effects are included in the shear force without interaction, τ_W. This application results in a slightly lower factor but has the advantage of permitting a step-by-step build up and comparison of the overall viscous forces. The magnitude of the skin-friction correction factor using the above techniques is shown in Figure 12.

The induced pressure on a surface is determined as an increment in pressure coefficient.

$$\Delta C_p = C_p - C_{p_o} = \frac{\bar{P} - P_o}{\frac{\gamma}{2} M^2}$$

The average pressure increment, $\bar{P} - P_o$, is found by summing the local pressure distribution over the surface.

$$\bar{P} - P_o = \frac{1}{A} \int (P - P_o)\, dA$$

Substituting the expression for local pressure and integrating gives

$$\bar{P} - P_o = 2m\lambda c_r$$

The ΔC_p due to induced pressure is determined for the skin-friction geometry representation of the vehicle shape and effects due to the planform shape and due to the initial surface are discussed in the next section.

The basic hypersonic small-disturbance relations for calculating pressure are:

For compression flow ($K \geq 0$)

$$P = 1 + \gamma\left(\frac{\gamma+1}{4}\right) K^2 + \gamma K \left\{1 + \left(\frac{\gamma+1}{4} K\right)^2\right\}^{1/2}$$

For expansion flow ($-2/(\gamma - 1) \leq K \leq 0$)

$$P = \left[1 + \frac{\gamma-1}{2} K\right]^{2\gamma/(\gamma-1)}$$

The similarity parameter, K, is given by;

$$K = K_o + \frac{\lambda K_4}{\sqrt{P}} \left[1 + \frac{\lambda}{2P} \frac{dp}{d\lambda}\right]$$

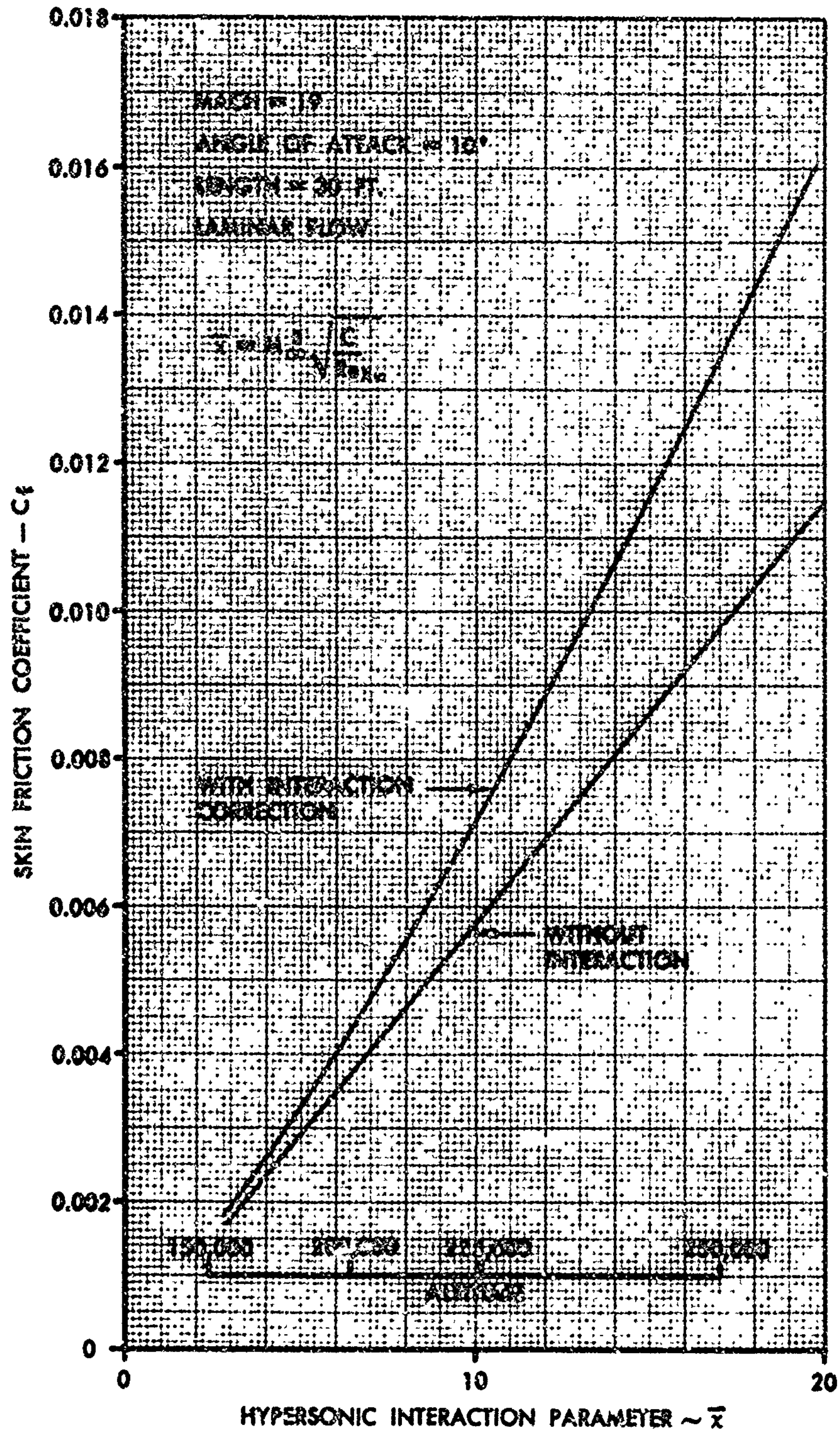


Figure 12. Effect of Viscous Interaction on Skin Friction Coefficient

where $K_o = M_\infty \sin\delta$ (δ is the surface impact angle) and K_4, a boundary-layer growth parameter, is taken equal to 1.0.

White (Reference 49) observed that the pressure equation (either compression or expansion) and the expression for K constituted a first-order nonlinear differential equation in $P(\lambda)$ and obtained numerical solutions directly without iteration. The results are shown in Table V from which White also observed that the pressure could be approximated by the linear relationship

$$P = P_o + m\lambda$$

where P_o and the slope parameter, m, are just functions of K_o. P_o is given by the hypersonic similarity relations as a function of K_o and, in the Arbitrary-Body Program, m is approximated to the data of Table V by the following analytical curves:

For $-2/(\gamma - 1) \leq K < -3.0$,

$$m = 1.424 + 0.219\, K_o$$

For $K \geq -3.0$,

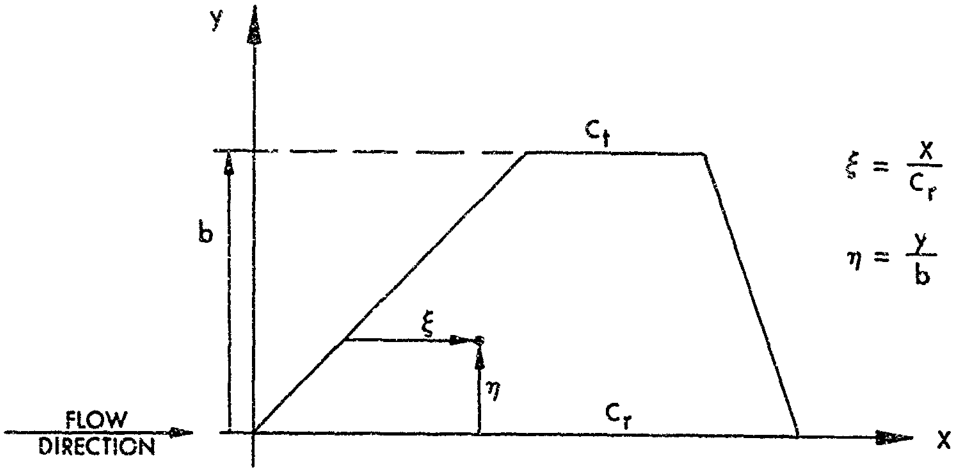
$$m = 1.9156 + 0.41727\, K_o - 0.0419101\, K_o^2 - 0.010427\, K_o^3 + 0.00214381\, K_o^4 - 0.000103217\, K_o^5$$

λ	Similarity Parameter K_o							
	-3	-2	-1	0	+1	+2	+5	+10
0.0	0.002	0.028	0.210	1.000	3.473	8.734	44.14	170.2
0.5	0.173	0.339	0.748	1.835	4.555	9.930	45.41	171.4
1.0	0.428	0.736	1.379	2.777	5.722	11.18	46.70	172.7
1.5	0.738	1.192	2.059	3.740	6.914	12.47	48.01	174.0
2.0	1.092	1.695	2.770	4.709	8.108	13.76	49.33	175.3
2.5	1.485	2.234	3.506	5.679	9.294	15.07	50.66	176.6
3.0	1.908	2.801	4.260	6.651	10.47	16.37	51.99	177.9
3.5	2.359	3.392	5.029	7.622	11.64	17.67	53.34	179.3
4.0	2.833	4.004	5.810	8.593	12.80	18.96	54.70	180.6
4.5	3.328	4.632	6.601	9.505	13.95	20.25	56.06	181.9
5.0	3.840	5.275	7.400	10.54	15.09	21.52	57.42	183.2

Table V. Numerical Solutions for Pressure Ratio P (γ = 1.4)

Planform Effects

The previous sections have dealt with the determination of the local skin-friction coefficient or the average skin-friction coefficient per unit span. In this section, the determination of the viscous force contribution of a surface element having a planform shape of the type shown in the sketch below is considered. In the derivations that follow it is implicitly assumed that the root and tip chords are parallel to the oncoming flow.



The product of local skin-friction coefficient (C_{f_δ}) and dynamic pressure (q_δ) is integrated over the surface area (A) to obtain the shear force:

$$\tau_W = \int q_\delta \, C_{f_\delta} \, dA$$

(The symbol τ is customarily used to define shear stress, however in the present text it is used consistently as a force. This is done to avoid the unnecessary use of area ratios in the defining equations and at the same time retain the significant connotation associated with the symbol.)

The shear force on each surface is then written as a coefficient with respect to the free-stream dynamic pressure (q_∞) and a specified reference area (S),

$$C_{F_\infty} = \frac{\tau_W}{\frac{1}{2} q_\infty S}$$

and summed over all surfaces to obtain the vehicle characteristics due to viscous forces.

Laminar Shear Force

The local properties are constant on each surface and the above expression becomes

$$\tau_W = q_\delta \; (C_{f_\delta})_{c_r} \; c_r \int_0^b \left\{ \int_0^c (x)^{-\frac{1}{2}} dx \right\} dy$$

where the surface has root chord c_r, span b, and $(C_{f_\delta})_{c_r}$ is evaluated at the root chord. The local chord length may be expressed as

$$c = c_r \left[1 - (1 - TR)\eta\right]$$

where TR is the taper ratio (= c_t/c_r) and η is the normalized span dimension (= y/b). Substituting this expression and completing the integration gives the shear force on the surface as

$$\tau_W = q_\delta \; A \, (C_{F_\delta})_{c_r} \; \frac{4}{3} \left[\frac{1 + TR + \sqrt{TR}}{(1 + TR)(1 + \sqrt{TR})}\right]$$

where $(C_{F_\delta})_{c_r}$ is the local, length-averaged skin-friction coefficient evaluated at the root-chord.

In the Arbitrary-Body Program the shear force is expressed in terms of an average chord length, $\bar{c}$;

$$\tau_W = q_\delta \; A \, (C_{F_\delta})_{\bar{c}}$$

where

$$\bar{c} = c_r \left\{ \frac{4}{3} \left[\frac{1 + TR + \sqrt{TR}}{(1 + TR)(1 + \sqrt{TR})}\right] \right\}^2$$

Viscous-Interaction

As was explained in the previous section, the effect of planform on the shear force is not determined directly for flows with viscous-interaction but is included in the calculation of shear force without interaction. This procedure results in a slightly lower force but has the advantage of permitting a step-by-step build-up and comparison of the overall viscous forces. There is, however, an additional effect on the induced pressure due to planform shape which is accounted for.

The average pressure is obtained by integrating the local pressure over the surface:

$$\bar{P}_A = \frac{1}{A}\int P\,dA = \frac{1}{A}\int_0^b \left\{ \int_0^c P\,dx \right\} dy$$

$$= \frac{c_r b}{A} \int_0^1 \left\{ \int^{c/c_r} (P_o + m\lambda c_r \xi^{-\frac{1}{2}}) d\xi \right\} d\eta$$

where $\xi = x/c_r$, the normalized streamwise coordinate.

Substituting the expressions for

$$A = \frac{c_r b}{2} (1 + TR)$$

and

$$c/c_r = 1 - (1 - TR)\eta$$

the integration is easily completed. The result is

$$\bar{P}_A = P_o \left\{ 1 + \frac{8}{3} B_{c_r} \left[\frac{1 + TR + \sqrt{TR}}{(1 + TR)(1 + \sqrt{TR})} \right] \right\}$$

where

$$B_{c_r} = \frac{m}{P_o} \lambda c_r$$

The average pressure increment for the surface is then

$$\bar{P}_A - P_o = \frac{8}{3} m\lambda_{c_r} \left[\frac{1 + TR + \sqrt{TR}}{(1 + TR)(1 + \sqrt{TR})} \right]$$

which for $TR = 1$ reduces to the value previously given.

Turbulent Shear Force

Because of the nature of the assumed skin-friction formulas, a different approach than used for laminar flow is taken to obtain the turbulent shear force. The end result, however, is an approximate solution which is very similar to the laminar result. The shear force equation is derived as follows.

$$\tau_W = \int q_\delta C_{f_\delta} \, dA = q_\delta \int_0^b \left\{ \int_0^c C_{f_\delta} \, dx \right\} dy$$

$$= q_\delta b \int_0^1 c \, C_{F_\delta} \, d\eta$$

The variable of integration is transformed to the local chord-length Reynolds number in two steps. First in terms of the chord length c,

$$\tau_W = q_\delta b \int_{c_t}^{c_r} \frac{c \, C_{F_\delta}}{c_r(1 - TR)} \, dc$$

Next, the variable of integration is transformed to the incompressible Reynolds number, $Rc_i = F_{Rx} \left(\frac{\rho U c}{\mu}\right)_\delta$, and normalized with respect to root-chord values;

$$\tau_W = \frac{q_\delta b \, c_r \, (C_{F_\delta})_{c_r}}{(1 - TR)} \int_{TR}^{1} \left(\frac{Rc}{Rc_r}\right)_i \left(\frac{C_F}{C_{Fc_r}}\right)_\delta d\left(\frac{Rc}{Rc_r}\right)_i$$

Noting that the surface area is $A = \frac{c_r b}{2}(1 + TR)$, and also that $\left(\frac{C_F}{C_{Fc_r}}\right)_\delta = \left(\frac{C_F}{C_{Fc_r}}\right)_i$, the shear equation becomes

$$\tau_W = q_\delta A (C_{F_\delta})_{c_r} \left(\frac{2}{1 - TR^2}\right) \int_{TR}^{1} \left(\frac{Rc}{Rc_r}\right)_i \left(\frac{C_F}{C_{Fc_r}}\right)_i d\left(\frac{Rc}{Rc_r}\right)_i$$

With a simple power-law skin-friction formula this equation is easily evaluated;

$$\left(\frac{C_F}{C_{Fc_r}}\right)_i = \left(\frac{Rc}{Rc_r}\right)_i^{-\frac{1}{N}}, \quad \text{where N is positive}$$

and

$$\tau_W = q_\delta A (C_{F_\delta})_{c_r} \left(\frac{2}{1 - TR^2}\right) \int_{TR}^{1} \left(\frac{Rc}{Rc_r}\right)^{1 - \frac{1}{N}} d\left(\frac{Rc}{Rc_r}\right)_i$$

$$= q_\delta A (C_{F_\delta})_{c_r} \left(\frac{2}{2 - \frac{1}{N}}\right) \left(\frac{1 - TR^{2 - \frac{1}{N}}}{1 - TR^2}\right)$$

For laminar flow $N = 2$ and it is easily verified that this expression is identical to the one previously presented.

In general, the skin-friction coefficient is not given by a simple power-law relationship and this is the reason for deriving the turbulent shear with the Reynolds number as the independent variable.

The use of the Sivells and Payne formula in the shear equation introduces a singularity in the integrand and the function is nonintegrable. However, this singularity occurs at a Reynolds number much below the laminar cutoff and the shear equation may be integrated numerically. Several examples for the numerically determined integrand are shown in Figure 13. The upper-bound represented by laminar flow and a lower-bound represented by constant skin-friction are also shown. The curves are smooth and the area under each curve times the quantity $2/(1 - TR^2)$ is the factor by which the shear increases due to a tapered planform.

It may be observed from Figure 13, that even with a large variation of Reynolds number on the planform (for example, $Rc_r = 10^9$ to zero at the tip), the major contribution to the integral is obtained over the first decade ($Rc/Rc_r = 1.0$ to 0.1). In the case of the upper-bound (laminar flow) and the lower-bound ($N = \infty$) this contribution is 97 and 99 percent, respectively. This then, suggested the approximate approach of representing the Sivells and Payne formula in the integrand over the entire Reynolds number range by a local power-law fit obtained as the average over the first decade.

Thus, the shear on the surface is obtained from the power-law solution with the exponent parameter, N, given as (for Sivells and Payne);

$$N = \frac{\log Rc_r - 2}{0.8686}$$

Alternately, as was done for laminar flow, the shear force may be expressed in terms of an average chord, $\bar{c}$;

$$\tau_W = q_\delta A(C_{F_\delta})_{\bar{c}}$$

where

$$\bar{c} = c_r \left(\frac{Rc_r}{10^{3/2}}\right)^{Q-1}$$

and

$$Q = \left\{ \left(\frac{1 - TR^2}{1 - TR^{2 - \frac{1}{N}}}\right)\left(\frac{2 - \frac{1}{N}}{2}\right) \right\}^{\frac{1}{2}}$$

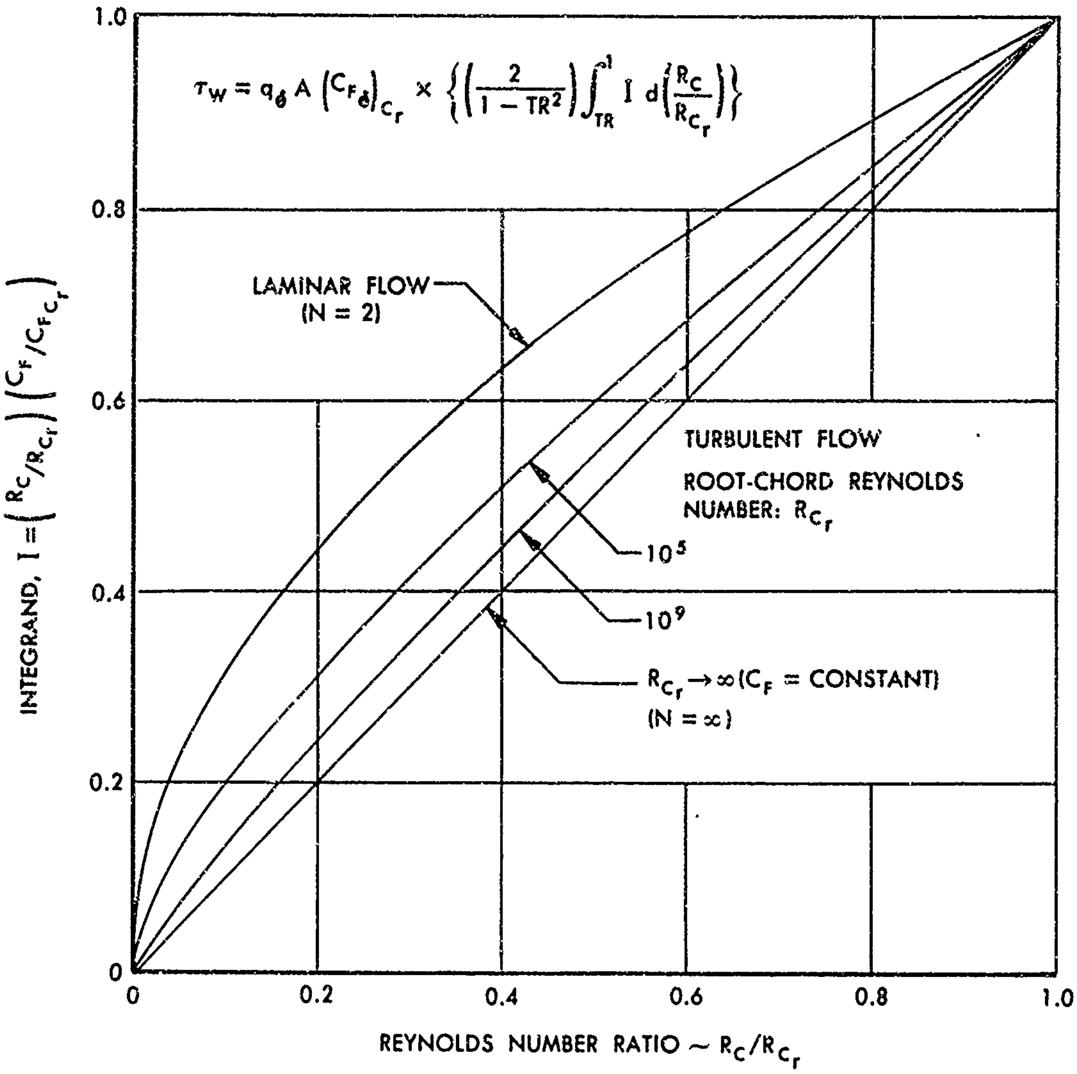


Figure 13. Planform Effect on Shear Force

Initial Surface Correction to Shear Force

When an initial surface is specified, the shear force is determined for the combined surface geometry, for the initial surface, and the difference obtained as the value for the surface of interest. This in effect is dealing with three surfaces which have the following characteristics (see sketch below):

1. Initial surface; Area A_1, maximum chord length L_1, taper ratio TR_1, and shear force τ_{W_1}.

2. Surface of interest; Area A_2, maximum chord length L_2, taper ratio TR_2, and shear force τ_{W_2}.

3. Combined surface; Area $A_3 = A_1 + A_2$, maximum chord length L_3, taper ratio TR_3, and shear force τ_{W_3}.

The shear force on surface 2 is

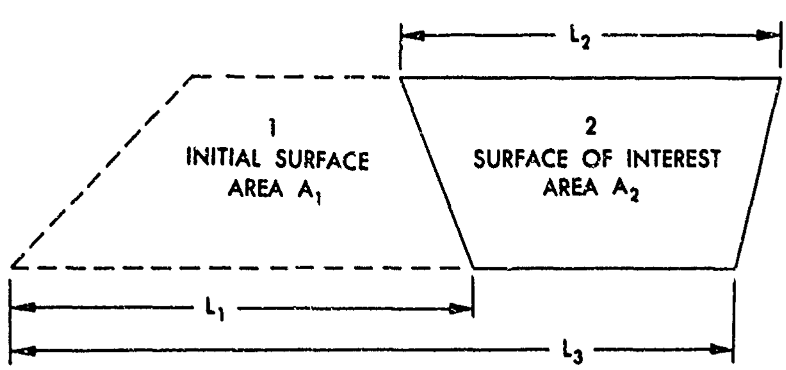
$$\tau_{W_2} = \tau_{W_3} - \tau_{W_1}$$

$$= q_\delta A_2 (C_{F_\delta} K_{VI})_3 \left\{ 1 - \frac{A_1}{A_2} \left[\frac{(C_F K_{VI})_1}{(C_F K_{VI})_3} - 1 \right] \right\}$$

In the Arbitrary-Body Program this is compacted to the form

$$\tau_{W_2} = q_\delta A_2 (C_{F_\delta} K_{VI})_3 \ (1 - FF)$$

where FF has the mnemonic form factor or friction factor. Three possibilities are considered in determining the friction factor: (1) both surfaces laminar, (2) first surface laminar and second surface turbulent, and (3) both surfaces turbulent.



Initial Surface Correction to Induced Pressure

The average pressure on surface 2 is defined as follows:

$$P_2 = \frac{F_2}{A_2} = \frac{F_3 - F_1}{A_2} = \frac{P_3 A_3 - P_1 A_1}{A_2}$$

where F_i is the force on surface i. The average pressures on the initial surface and on the combined surface are given by

$$P_1 = P_o \left\{ 1 + \frac{8}{3} B_1 \left[\frac{1 + TR_1 + \sqrt{TR_1}}{(1 + TR_1)(1 + \sqrt{TR_1})} \right] \right\}$$

$$P_3 = P_o \left\{ 1 + \frac{8}{3} B_3 \left[\frac{1 + TR_3 + \sqrt{TR_3}}{(1 + TR_3)(1 + \sqrt{TR_3})} \right] \right\}$$

and the areas by

$$A_1 = bL_1 (1 + TR_1)/2$$

$$A_2 = bL_2 (1 + TR_2)/2$$

$$A_3 = bL_3 (1 + TR_3)/2$$

Substituting these expressions into the above definition and after some algebraic manipulation the result may be written as

$$\overline{P}_2 - P_o = \frac{8}{3} m\lambda_3 \left(\frac{L_3}{L_2}\right) \left[\frac{1+TR_3 + \sqrt{TR_3}}{(1+TR_2)(1+\sqrt{TR_3})} \right] \left\{ 1 - \left(\frac{L_1}{L_3}\right)^{\frac{1}{2}} \left(\frac{1+TR_1+\sqrt{TR_1}}{1+TR_3+\sqrt{TR_3}}\right) \left(\frac{1+\sqrt{TR_3}}{1+\sqrt{TR_1}}\right) \right\}$$

The length L_3 is defined as the maximum chord length of the combined surface, so as $L_1 \rightarrow 0$ it is readily verified that the pressure reduces to the same expression previously given for a single, tapered plate.

Viscous Force on Blunt Bodies

The earliest space capsules were designed with large spherical nose caps and flew ballistically at zero degrees angle of attack. For such vehicles, it was found that inviscid flow field calculations were adequate to predict the splash point. The later generation capsules were designed to fly at angle of attack to provide lift and it has been shown that viscous forces can have a significant effect on predicting the splash point. The theoretical solution, then, must provide some means for estimating the viscous effect.

The procedure included in the Arbitrary-Body Program is that developed by Goldberg of the General Electric Company (References 52 and 53). This method is given in the form of relatively simple correlation formulas in terms of the shock-layer Reynolds number and inverse density ratio. The method is applicable to the low density conditions associated with high altitude entry and is equally suited to real gas or ideal gas analysis.

The shear force in the stagnation region of a blunt-faced body is given as

$$\tau_W = \tau_{W_o} K_{VI}$$

where the shear without low density or viscous-interation effects is

$$\tau_{W_o} = \frac{q_\infty A\ 2 \cos \delta}{(1 - 0.495\sqrt{\epsilon})\sqrt{R_{e_s}}}$$

and δ is the surface impact angle,

ϵ is the inverse density ratio, $= (\rho_2/\rho_\infty)^{-1}$

R_{e_s} is the shock Reynolds number

$= \rho_2 U_2 R_B/\mu_2$

R_B is the body nose radius.

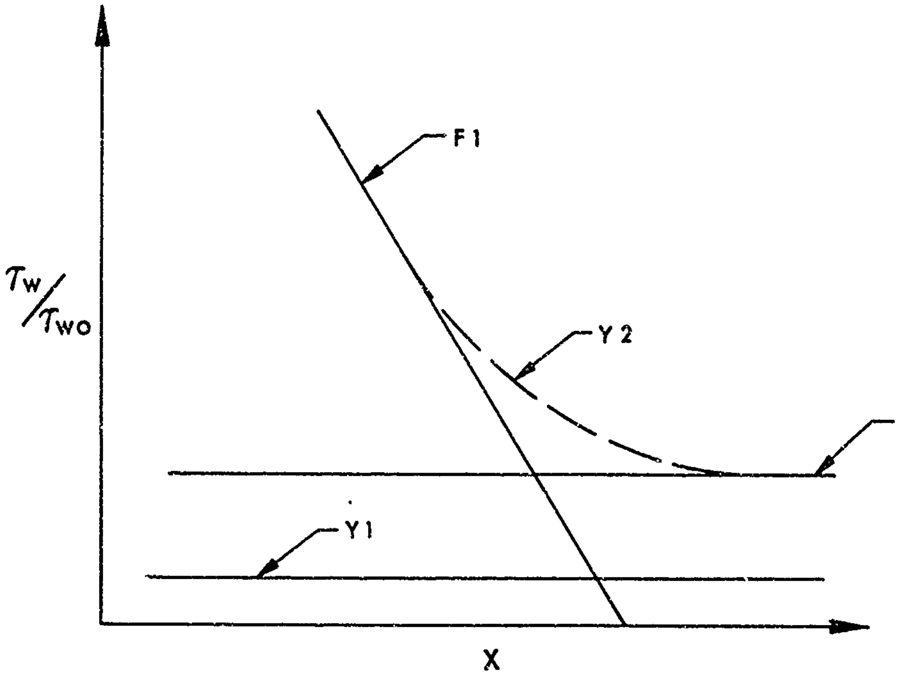
The viscous-interaction correction factor, $K_{VI} = \tau_W/\tau_{W_o}$, was obtained from higher-order analysis of the boundary-layer flow (Reference 52). The present authors have developed a correlation formula to represent these solutions in the Arbitrary-Body Program. This factor, a complicated function of both shock Reynolds number and density ratio, has been approximated by a combination of exponential transition functions of the type described by Grabau (Reference 54). These are

$$\text{even transition:} \quad y = \frac{1}{1 - \exp K(X - X_o)}$$

$$\text{odd transition:} \quad y = \frac{1}{1 + \exp K(X - X_o)}$$

These functions are essentially the kernels for the Bose-Einstein and for the Fermi-Dirac distribution functions, respectively, for the even and the odd transitions. The notation of transition is used since these functions represent the smooth transition from one asymptote to another; the even case does not have a point of inflection and the odd transition has a point of inflection.

In the present application, a correlation formula for the viscous-interaction parameter has been obtained by a combination of an even and odd transition function. The curve is considered to have three asymptotes (see the sketch below); Y1, F1, and F2. First an even transition is determined for the curve between F1 and F2 and this is designated Y2. Next, an odd transition is established between Y1 and Y2. The curves are adjusted through the values specified for the exponential constants, K, and the origin coordinates, X_o. Details of this procedure are given in Reference 54.



The correlation formulas developed for the present case are as follows.

Independent variable $X = \log(\epsilon^3 R_{e_s})$

$$F1 = A1 + B1(X)$$

$$A1 = 0.667$$

$$B1 = 1.1111$$

$$F2 = 1.0$$

$$Y1 = 0.0$$

$$Y2 = F1 + \frac{(1.0 - F1)}{1.0 - \exp\left[EVK\,(X - XOEV)\right]}$$

$$EVK = -1.80$$

$$XOEV = -0.3$$

$$\tau_W / \tau_{W_b} = \frac{Y2}{1.0 + \exp\left[ODK\,(X - XOOD)\right]}$$

$$ODK = -2.0$$

$$XOOD = AOD + BOD(\log \epsilon)$$

$$AOD = 1.0$$

$$BOD = 3.2907$$

Comparison of this correlation and the boundary-layer solutions are shown in Figure 14. The general shape of the curves is well represented by the correlation, although some accuracy is lost, particularily at the peak of the $\epsilon = 0.04$ curve. It would be possible to tailor-fit each of the ϵ-curves through further variation in F1, the exponential constants and origin coordinates. However, since only three solutions were available, the determination of more accurate fits was not deemed justified. Three additional ϵ-curves are given on the figure to demonstrate the behavior of the correlation formula.

An example of this technique is shown in Figure 15 were the predicted values of lift coefficient for the Gemini space capsule are compared with experimental results (Reference 55). The modified Newtonian calculation has been performed for the entire shape and the viscous calculations (broken lines) made only for the blunt face. The present comparison, due to the limited data used, may not completely justify the method, but it does show the significance of the viscous contributions.

The blunt-body viscous calculations are not limited to entry capsules but may be applied to any blunt portions of a vehicle (e.g., leading edges). The method is primarily dependent on impact angle and, therefore, the detailed inviscid geometry is used. It is for this reason that the method has been included as one of the inviscid force options. Zero contribution is assumed for shadow flow.

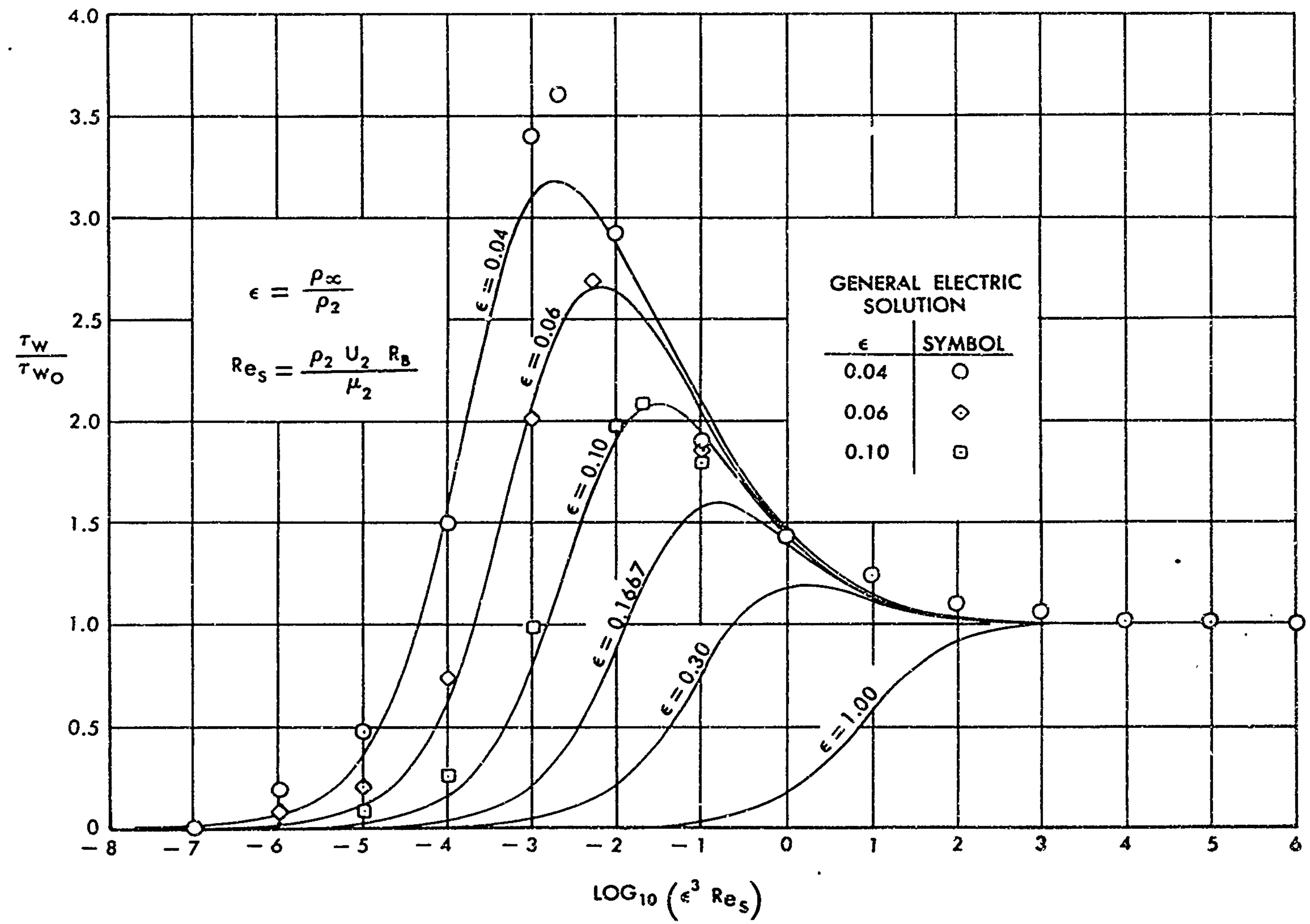


Figure 14. Low Density Correction to Blunt-Body Viscous Forces

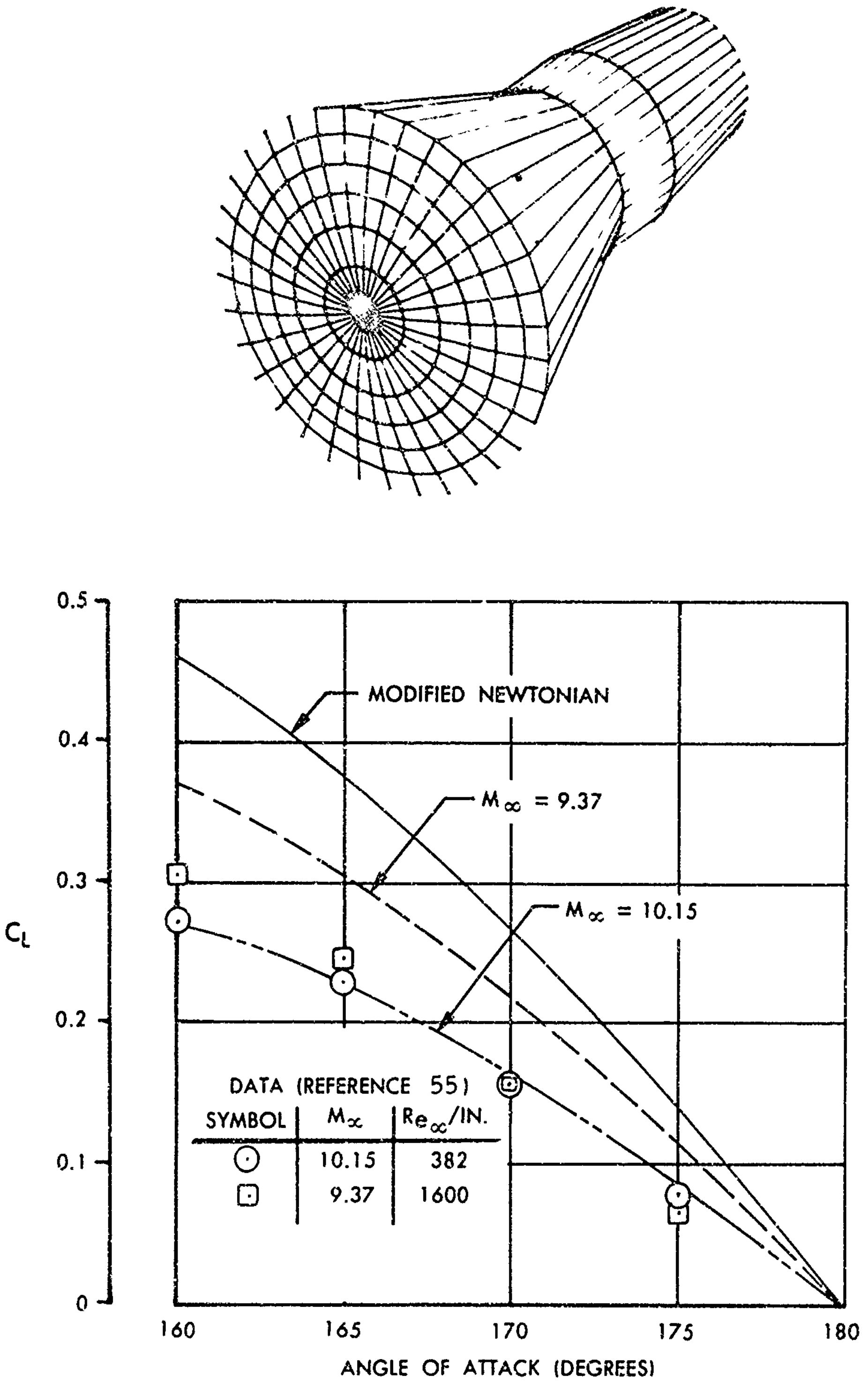


Figure 15. Gemini Lift Coefficient Comparison

Control Surface Forces

An important feature of the flow about hypersonic control surfaces is the boundary-layer flow separation phenomenon. Flow separation on the control surface and on the surface of the vehicle ahead of the control can have a pronounced influence on control effectiveness. This is a very difficult problem to analyze theoretically with any degree of accuracy. However, the use of a simplified flow model and empirical boundary-layer separation data will allow the solution of this problem with sufficient accuracy for most preliminary design purposes.

The basic factors involved in the control surface flow separation phenomena are illustrated in Figure 16. In these flow models the pressure in the separated region is taken to be the plateau pressure. This region is well defined for laminar flow and may affect a large portion of the vehicle surface on, and ahead of, the control flap. For turbulent flow the separated region is much smaller but, although a true pressure plateau may not exist, a flow model using the inflection pressure will produce useful results for most purposes.

The boundary layer separation at point X_{SEP} is termed a free-separation and the separation angle in this flow model depends only on the local Mach number (M_0) and Reynolds number (R_{x_0}) ahead of the separation region. The extent of the separation region as indicated by the separation length ℓ_{SEP} is determined by the strength of the shock at the flow reattachment point X_R. As the control surface deflection angle $_f$ is increased, the flow turning angle at the reattachment point increases, the final overall pressure rise after the reattachment shock increases, more flow is forced back up the boundary layer and into the separation region, and the length of the separation region increases to accommodate the greater flow in the separated area. The length of the separation region continues to increase until the reattachment point reaches the control surface trailing edge.

Present theoretical methods are completely inadequate for the prediction of separation effects; complete reliance, therefore, must be placed on empirical correlations. The relationships used in this program were taken from the work of Popinski and Ehrlich in Reference 56. The basic approach used in the Arbitrary-Body Program is essentially the same as this reference. There are, however, a few significant differences in the details of the program application. Since the program can perform shock-expansion calculations this capability has been exploited in the flow-separation application. Also, since all of the calculations are carried out by the computer program rather than being done by hand it is possible to use a strip theory approach that permits a rough assessment of the three-dimensional effects. The computations within the program assume an ideal gas.

The application of the flow-separation criteria in this program also has another very significant different from Reference 56. The program uses shock-expansion theory to determine the flow separation pressure changes and then applies these separation corrections to the pressures calculated for the vehicle, using any of the other force methods. This process is illustrated in Figure 17.

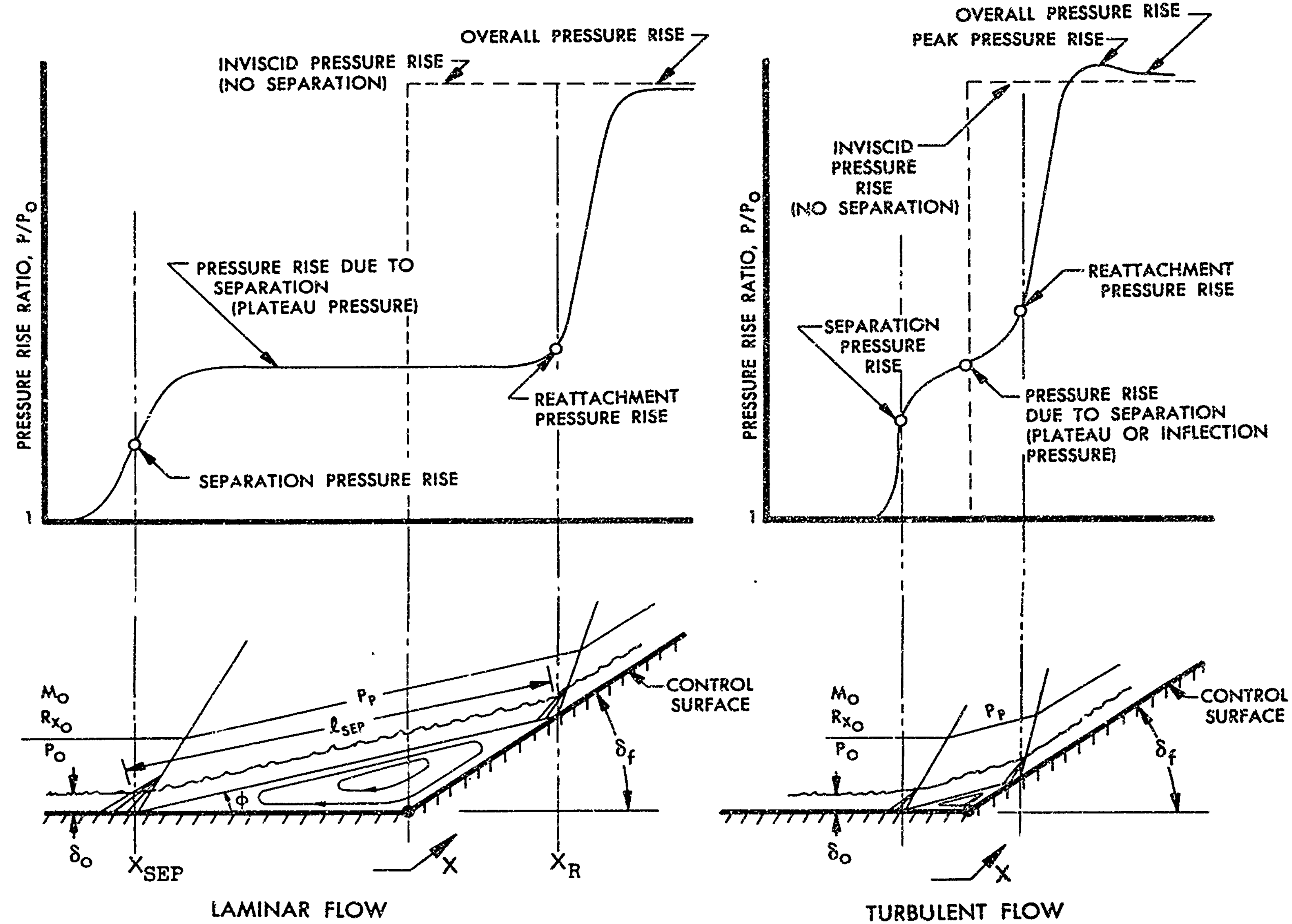


Figure 16. Wall Pressure Distribution in the Vicinity of Separation

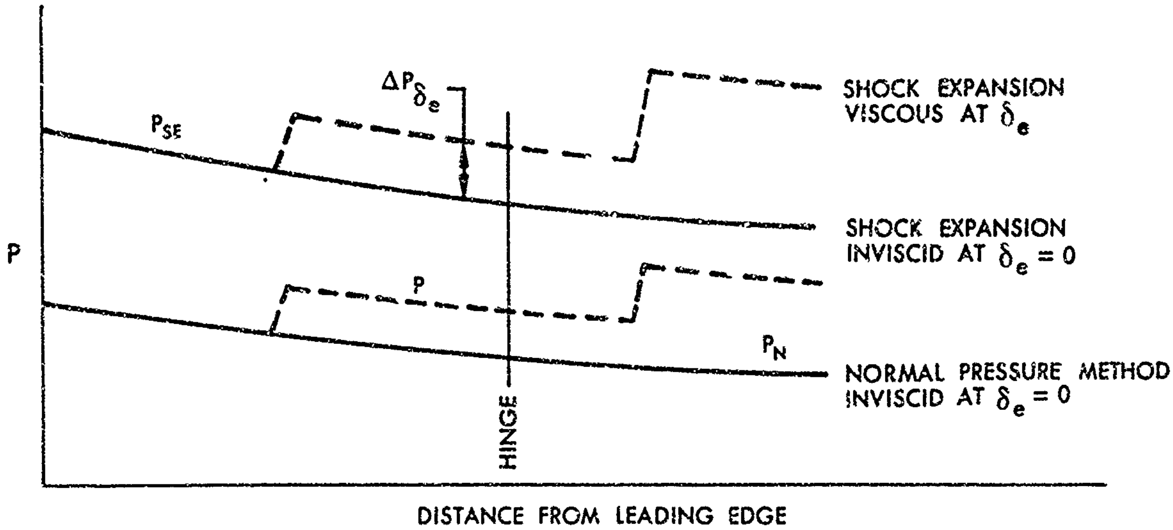


Figure 17. Correction of Normal Pressure Method Results for Separation Effects

The final surface pressure is calculated by the relationship shown below.

$$P = P_N + \left(\frac{\Delta P_{\delta_e}}{P_{SE}}\right) \left(P_N\right)$$

This technique not only gives a smooth trend of data as the deflection angle goes to zero, but also insures that it will match the basic vehicle results using whatever normal force method has been selected.

The essential features of the flow-separation calculations are given below. The notion is maintained consistent with Reference 56.

The calculation of flow-separation effects uses a three-cycle process. On the first cycle a streamwise strip of elements is analyzed using the shock-expansion method to determine local conditions at the hinge point. The inviscid pressure rise onto the flap is calculated also and compared with the incipient separation criteria. The transition Reynolds number is input to the program.

On the next cycle the shock-expansion method is again applied to the same streamwise strip of elements, only this time separation effects are accounted for. This is accomplished by checking at each element to see if the flow separation point has been reached, and by calculating the new pressures. Again, the equations in Reference 56 were used in these computations. For other reviews of the separation problem see References 57, 58, and 59.

On the third cycle of calculations along the streamwise strip of elements the pressures are calculated using the normal input pressure calculation method (IMPACT and ISHAD). At the same time, these pressures are corrected by the flap deflection and flow separation pressure increments.

The above discussion gave a brief outline of the techniques involved in the flow separation calculations. Of course, a thorough study of the listing of the FLOSEP subroutine is necessary to obtain a thorough understanding of the methods used. The following description of the equations used will assist in this study.

The basic purpose of the first flow separation analysis cycle is to determine the local flow properties at the hinge-line element, and with this information, to determine the pressure rise in going from the fore-surface element at the hinge line to the first element on the flap. This pressure rise is then compared with the empirical incipient pressure rise equation to determine if the flow separates.

On the first FLOSEP cycle the flow properties on each element in the streamwise strip are calculated by the shock-expansion method. This gives complete freedom to analyze curved vehicle surfaces in front of a flap (the fore-surface) and also on the flap itself. In many such cases this approach will give useful results even though the empirical separation criteria equations used are derived from tests on flat surfaces only.

When the hinge-line element is reached on the first FLOSEP cycle the program checks for flow separation. The flow turning angle at the hinge line caused by the control deflection is calculated from the following relationship.

$$\delta_f = \sqrt{(N_Y N_{Z_H} - N_Z N_{Y_H})^2 + (N_Z N_{X_H} - N_X N_{Z_H})^2 + (N_X N_{Y_H} - N_Y N_{X_H})^2}$$

where

N_X, N_Y, N_Z = the direction cosines of the first element on the flap.

$N_{X_H}, N_{Y_H}, N_{Z_H}$ = the direction cosines of the hinge-line element.

Note that when the flap is not a plane surface the flow deflection angle calculated by the above equation will be different than the program input control surface deflection angle.

The inviscid pressure rise at the flap is calculated by the shock-expansion method (oblique shock or Prandtl-Meyer expansion as the case may be). This pressure rise is then compared with the incipient pressure rise criteria given in Reference 56.

$$(C_{P_{\alpha}})_{inc} = \frac{2.03\,(M_\alpha^2 - 1)^{-0.306}}{(R_{e_{\alpha_{HL}}})^{1/4}} \qquad \text{Laminar}$$

$$(C_{P_{\alpha}})_{inc} = \frac{2.2}{(R_{e_{\alpha_{HL}}})^{1/10}} \qquad \text{Turbulent}$$

where

$(C_{P_{\alpha\, inc}})$ = the pressure rise required to cause incipient separation.

M_α = the local Mach number at the hinge line.

$R_{e_{\alpha_{HL}}}$ = the Reynolds number at the hinge line based on the local hinge-line flow properties and the distance from the leading edge.

Subsequent FLOSEP calculation cycles require a value for the flap chord length (C_{flap}). This length is taken as being the distance from the hinge line to the flap trailing edge as given by the following equation.

$$C_{flap} = \sqrt{(X_{TE} - X_{HL})^2 + (Y_{TE} - Y_{HL})^2 + (Z_{TE} - Z_{HL})^2}$$

where

X_{TE}, Y_{TE}, Z_{TE} = the average values of the X, Y, Z element coordinates at the trailing edge.

Viscous separation effects are calculated on the second FLOSEP calculation cycle. The general procedure used differs from the method of Reference 56 in that no iteration process is used to determine the separation point. Since the vehicle surface is not restricted to flat surfaces this technique is not possible. Instead, as the local flow properties on each stream-wise element are calculated a check is made with the appropriate equations to see if the separation point has been reached. When flow separation is indicated on a given element a linear interpolation of parameters between the last non-separated element and the separated-element is used to determine the exact point and conditions at the separation.

The following calculations are required to provide information for the check for flow separation. The local Reynolds number at the element centroid is given by

$$R_{e_{\alpha X_o'}} = X_{LE}\ (R_{e_\alpha} / ft)$$

where

X_{LE} = the distance from the leading edge to the element centroid.

R_{e_α} / ft = the Reynolds number per foot based on the local element flow conditions.

The plateau pressure rise due to separation is given by the following equations.

$$(C_{P_\alpha})_P = \frac{1.56\,(M_\alpha^2 - 1)^{-0.262}}{(R_{e_\alpha X_o'})^{1/4}} \qquad \text{Laminar flow}$$

$$(C_{P_\alpha})_P = \frac{1.91\,(M_\alpha^2 - 1)^{-0.309}}{(R_{e_\alpha X_o'})^{0.1}} \qquad \text{Turbulent flow}$$

The ratio of the plateau pressure to the local element pressure upstream of the separation is given by

$$P_P/P_o = 0.7\,M_\alpha^2\,(C_{P_\alpha})_P + 1.0$$

The ratio of the distance from the start of separation to the hinge line (d_1), and the local boundary layer thickness (δ_o) is given by

$$d_1/\delta_o = 5.69 \times 10^5\,M_\alpha^{-4.1}\,(P_P/P_o - 1)^{3.5} \qquad \text{Laminar flow}$$

$$d_1/\delta_o = 1.1 \times 10^6 \left[M_\alpha^{-1.67}\,(P_P/P_o - 1)\right]^{8.55} \qquad \text{Turbulent flow}$$

The calculation of the boundary layer thickness requires the local surface Reynolds number per unit length based on the reference condition ($R_{e_\alpha}^*/\mathrm{ft}$). This parameter is obtained by the temperature subroutine (TEMP) used in the skin friction calculations. The surface wall temperature required to find the Reynolds number may either be determined by the temperature routine or input. The same options used in the viscous force calculations are available to FLOSEP.

The boundary layer thickness is given by the following equations.

$$\frac{\delta_o}{\sqrt{X_{LE}}} = \frac{5.2}{\sqrt{R_{e_\alpha}^*/\mathrm{ft}}} \qquad \text{Laminar}$$

$$\frac{\delta_o}{X_{LE}^{6/7}} = \frac{0.154}{(R_{e_\alpha}^*/\mathrm{ft})^{1/7}} \qquad \text{Turbulent}$$

From the above equations for d_1/δ_o and δ_o/X_{LE} we may obtain the distance from the start of separation to the hinge line (d_1). The separation distance parameters are shown in Figure 18 as taken directly from Reference 56.

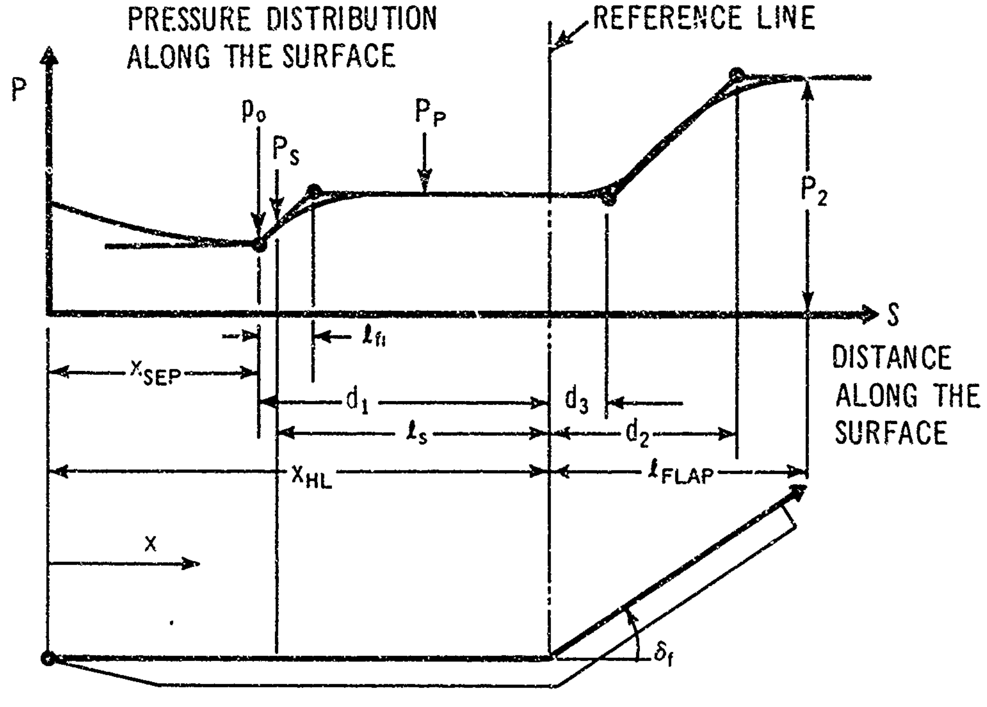


Figure 18. Definition of Interaction Parameters for Separated Flow

The check to see if the flow separation point has been reached is accomplished as follows.

$$X_{SEP} = X_{LE_H} - d_1$$

$$\Delta X_{SEP} = X_{LE} - X_{SEP}$$

If $X_{LE} \geq X_{SEP}$, then the separation point has been reached.

where

X_{LE_H} = the distance from the leading edge to the hinge line.

X_{SEP} = the distance from the leading edge to the separation point.

The exact flow separation point and the attendant flow properties are calculated by a linear interpolation between the data on the element before the separation and the actual separation element.

The procedure used in calculating the other required flow separation geometric parameters and the separation pressures is outlined below.

The ratio of the free interaction length (ℓ_{fi}) to the boundary layer thickness (δ_o) is obtained from the following equation.

$$\frac{\ell_{fi}}{\delta_o} = 2.47 \times 10^5 \; M_\alpha^{-4.2} \; (P_P/P_o - 1)^{3.45} \qquad \text{Laminar}$$

$$\frac{\ell_{fi}}{\delta_o} = 1.84 \times 10^4 \left[\frac{P_P/P_o - 1}{M_\alpha^{1.325}} \right]^{-8.4} \qquad \text{Turbulent}$$

The free interaction length is now calculated.

$$\ell_{fi} = (\ell_{fi}/\delta_o)\; \delta_o$$

If the flow is separated on the first element of a strip the parameter ℓ_{fi} is set equal to 0.0. Also, the parameter $X_{LE_{SEP}}$ is set equal to 0.0.

The ratio of the downstream interaction length to peak pressure (d_2), to the upstream interaction length (d_1) is given by the following equations.

For Laminar Flow

$$\frac{C_{flap}}{d_1} \geq 1 \; : \; \frac{d_2}{d_1} = .545 - .04\,(M_\alpha \delta_f)$$

$$\frac{C_{flap}}{d_1} \leq 0.25 \; : \; \frac{d_2}{d_1} = .273 - .02\,(M_\alpha \delta_f)$$

$$M_\alpha \delta_f \geq 5 \; : \; \frac{d_2/d_1}{\sqrt{\Gamma}} = .344$$

where

$$\Gamma = \begin{cases} \dfrac{C_{flap}}{d_1}, \text{ if } .25 \le \dfrac{C_{flap}}{d_1} \le 1 \\ 1, \text{ if } \dfrac{C_{flap}}{d_1} \ge 1 \\ .25, \text{ if } \dfrac{C_{flap}}{d_1} \le .25 \end{cases}$$

For Turbulent Flow

$$\frac{C_{flap}}{d_1} \ge 1 \; : \; d_2/d_1 = 1.16 - 0.33 \; M_\alpha \delta_f$$

$$\frac{C_{flap}}{d_1} \le 0.25 \; : \; d_2/d_1 = 0.58 - 0.165 \; M_\alpha \delta_f$$

$$M_\alpha \delta_f \ge 2.4 \; : \; d_2/d_1 \; \frac{1}{\sqrt{\Gamma}} = 0.37$$

where

$$\Gamma = \begin{cases} 1, \text{ for } \dfrac{C_{flap}}{d_1} \ge 1 \\ \dfrac{C_{flap}}{d_1}, \text{ for } 1 \ge \dfrac{C_{flap}}{d_1} \ge 0.25 \\ 0.25, \text{ for } \dfrac{C_{flap}}{d_1} \le 0.25 \end{cases}$$

From the above equations the downstream interaction length to the peak pressure (d_2) may now be calculated.

The angles associated with the separation are given by

$$\theta = \sin^{-1} \left[(C_{P_\alpha})_P \frac{\gamma + 1}{4} + \frac{1}{M_\alpha^2} \right]$$

$$\tan \phi = 1 \Bigg/ \left[\left(\frac{2.0}{(C_{P_\alpha})_P} - 1.0 \right) \frac{\sin \theta}{\cos \theta} \right]$$

where

ϕ = the flow deflection caused by separation.

θ = shock angle for the flow turning angle, ϕ.

If the flow has separated at the leading element the angles are given by

$$\phi = \tan^{-1}\left[\frac{d_2 \sin \delta_f}{d_1 + d_2 \cos \delta_f}\right]$$

The ratio of the downstream interaction length to pressure rise (d_3), to the upstream interaction length (d_1) is given by

$$\frac{d_3}{d_1} = \left(\frac{\tan \phi}{\tan \delta_f - \tan \phi}\right) \frac{1}{\cos \delta_f}$$

and

$$d_3 = (d_3/d_1)\, d_1$$

If the distance d_3 is greater than the flap chord length, C_{flap}, the separated region as determined from the above equation extends beyond the end of the flap. When this condition occurs the distance d_3 is set equal to the flap chord, C_{flap}, and a new value for d_1 calculated based on conditions that existed at the originally calculated separation point.

$$d_1 = d_3/(d_3/d_1)$$

This value for d_1 will be smaller than the original value and will give a new flow geometry model with the same flow turning angle, ϕ, but with the separated region reduced to extend to the exact flap trailing edge point.

The plateau pressure caused by the separation turning angle ϕ, ($C_{P_{I_P}}$), is calculated by the oblique shock compression subroutine.

The flow turning angle in going from the plateau pressure region to the final peak flap pressure area is found from

$$\phi_2 = \delta_f - \phi$$

The resulting peak flap pressure ($C_{P_{I_2}}$) is also calculated by the oblique shock compression subroutine.

The flow separation pressures are distributed on the fore-surface and the flap by using the following equations.

If $X_{LE} \leq X_{LE_{SEP}}$ $\qquad C_{P_{SEP}} = C_{P_{I_P}}$

If $X_{LE} \geq X_{LE_{SEP}}$ and $< (X_{LE_{SEP}} + \ell_{fi})$

$$C_{P_{SEP}} = \left(\frac{C_{P_{I_P}} - C_{P_X}}{\ell_{fi}}\right) X_{LE} + C_{P_X} - \left(\frac{C_{P_{I_P}} - C_{P_X}}{\ell_{fi}}\right) X_{LE_{SEP}}$$

where

C_{P_X} = the surface pressure at the separation point.

The following equations are used to distribute the separation pressures on the flap.

If $X_{LE} < (X_{LE_H} + d_2)$ $\qquad C_{P_{SEP}} = C_{P_{I_P}}$

If $X_{LE} > (X_{LE_H} + d_3)$ and $< (X_{LE_H} + d_2)$

$$C_{P_{SEP}} = \left(\frac{C_{P_{I_2}} - C_{P_{I_P}}}{d_2 - d_3}\right) (X_{LE} - X_{LE_H} - d_3) + C_{P_{I_P}}$$

For $d_2 > C_{flap}$ $\qquad C_{P_{SEP}} = C_{P_{I_P}}$

If $d_3 > C_{flap}$ $\qquad C_{P_{SEP}} = C_{P_{I_P}}$

If $X_{LE} \geq (X_{LE_H} + d_2)$ $\qquad C_{P_{SEP}} = C_{P_{I_2}}$

At the end of the second flow separation cycle the above pressure coefficients are converted to pressure with the following equation.

$$P_{SE_{vis}} = \left[C_{P_{SEP}} \frac{\gamma}{2} M_\infty^2 + 1.0\right] P_\infty$$

where

P_∞ = the free stream pressure.

M_∞ = the free stream Mach number.

$P_{SE_{vis}}$ = the pressure on the surface using shock-expansion methods and including viscous separation effects.

The purpose of the third cycle of calculations down a strip of elements is to determine the basic surface pressures using the normal program pressure calculation method (i. e., Newtonian, tangent-cone, etc.), and to determine the final surface pressures including the viscous separation effects. For this cycle the control surface is set to the undeflected position. The final surface pressure is given by

$$P = P_N + \left(\frac{\Delta P_{\delta_e}}{P_{SE}}\right) P_N$$

where

P = the final surface pressure including control surface effects.

P_N = the surface pressure calculated by the normal program input pressure calculation method.

ΔP_{δ_e} = the change in surface pressure due to control surface deflection and including separation effects.

P_{SE} = the surface pressure using shock-expansion methods without the separation effects.

Typical results obtained with the above analysis techniques are shown in Figure 19.

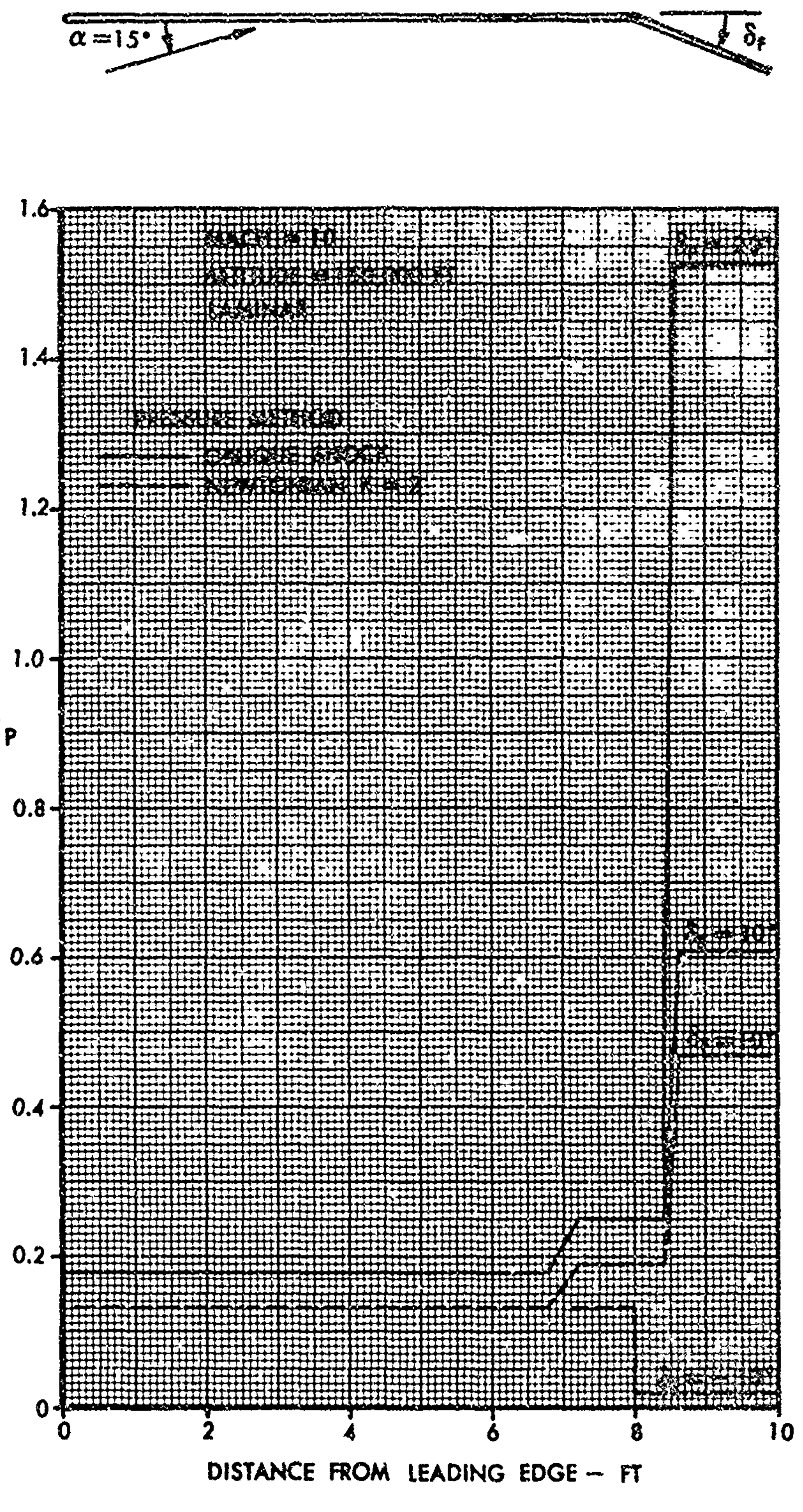


Figure 19. Effect of Flow Separation on Surface Pressure

Propulsion Effects

The operation and design of an air-breathing propulsion system of a hypersonic vehicle can have a strong influence on the vehicle stability and control characteristics. For some aircraft, such as scramjet-powered vehicles, the engine, including the inlet and exhaust systems, may be highly integrated into the vehicle design. For these configurations the bookkeeping system as to what is engine and what is airplane is no longer easy to resolve. A detailed analysis of these problem areas is obviously beyond the scope of this program. These problems must be solved by detailed engine-airframe studies.

It is important, however, that the Arbitrary-Body Program be capable of properly using the results from these propulsion system studies in evaluating a total vehicle's stability characteristics. The approach used is outlined in the following discussion.

In investigating the propulsion effects on hypersonic stability it is usually desirable to examine each of the various components of the engine thrust system. The equations for a general air-breathing propulsion system are given below.

$$T = \dot{m}_a (V_2) + \dot{m}_f V_2 + (p_2 - p_o) A_2$$

$$R = \dot{m}_a (V_1) + (p_1 - p_o) A_1$$

where

T = gross thrust

R = ram drag

$\dot{m}_a$ = mass flow of air through the engine system

$\dot{m}_f$ = mass flow of fuel added to the stream

V_1 = flow velocity at the forward flow control boundary

V_2 = flow velocity at the rearward (exhaust) flow control boundary

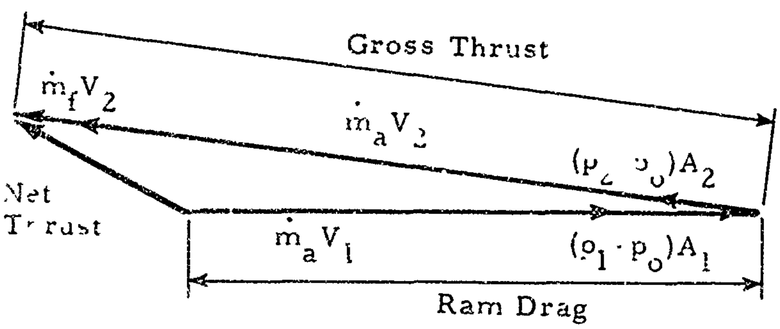
p_o = reference pressure

p_1 = pressure at control boundary 1

p_2 = pressure at control boundary 2

A = area

The vector relationships for a simple engine system are illustrated in the diagram below



The propulsion system has two main influences on a vehicle's stability characteristics. First, the engine ram drag and gross thrust vectors create moments about the vehicle center of gravity. Second, the exhaust jet may at times expand onto the aft surfaces of the vehicle and alter the local pressure distributions. The numerical values of the parameters involved depend upon the engine-airframe bookkeeping system adopted. The final result, however, should be the same under any system. Any given bookkeeping system serves only to insure that all factors are properly included and that no factors are included twice.

The engine related forces are input to the program in the form of vectors positioned in space. The input information includes the magnitude of the force in pounds, its point of action on the vehicle (in the vehicle X, Y, Z coordinate system), and its direction. The direction is given by the force vector direction cosines. Any number of vectors may be used.

The force vectors are converted into vehicle coefficients by the following equations.

$$\Delta C_A = F(-N_X)/(qS_{ref})$$

$$\Delta C_Y = F(-N_Y)/(qS_{ref})$$

$$\Delta C_N \quad F(N_Z)/(qS_{ref})$$

$$\Delta C_\ell \quad \Delta C_Y(Z_{cent} - Z_{cg})/SPAN$$

$$+ \Delta C_N(Y_{cent} - Z_{cg})/SPAN$$

$$\Delta C_M = \Delta C_N (X_{cent} - X_{cg})/MAC$$

$$+ \Delta C_A (Z_{cent} - Z_{cg})/MAC$$

$$\Delta C_N = \Delta C_Y (X_{cent} - X_{cg})/SPAN$$

$$- \Delta C_A (Y_{cent} - Y_{cg})/SPAN$$

where

F = force vector in pounds

N_X, N_Y, N_Z are the direction cosines of the force vector

q = free stream dynamic pressure

These coefficients are converted to the lift and drag directions by the standard transformation equations previously discussed.

Dynamic Stability Derivatives

The dynamic stability derivatives due to pitching velocity (C_{m_q}) and vertical acceleration ($C_{m_{\dot{\alpha}}}$) are important in the damping of the short-period stability mode. For flight at hypersonic speeds the derivative C_{m_q} provides the major part of the damping since $C_{m_{\dot{\alpha}}}$ is usually quite small.

The derivative C_{m_q} is the change in pitching moment coefficient with varying pitch velocity and is commonly referred to as the pitch damping derivative. This derivative represents the effect of rotation of the vehicle about a spanwise axis at constant angle of attack such as in a steady pull-up. Because the vehicle is rotating, different parts of the vehicle have different velocities relative to the free-stream depending upon their distance from the center of rotation. This is essentially a steady-state problem and may be solved by the use of steady flow concepts. The common definition for C_{m_q} is

$$C_{m_q} = \frac{\partial C_m}{\partial\left(\frac{q\bar{c}}{2V}\right)} .$$

where

C_m = pitching moment coefficient

q = pitching rate

$\bar{c}$ = reference chord for moment coefficients (frequently the mean aerodynamic chord)

V = flight velocity

The method used in the Arbitrary Body Program to calculate this derivative has been presented earlier in this report and will not be repeated again in this section. The basic approach used involves the calculation of the local relative flow velocity for each part of the vehicle depending upon its distance from the center of rotation and the rotation rate. Once the local velocity direction is known the surface pressure is calculated using any of the methods available in the progra. as desired (except the shock-expansion method).

The results of the rotational derivative calculations depends upon the amount of detail used in the geometrical description of a shape. This effect is illustrated in Figures 20 through 23. These figures contain the results of the C_{m_q} calculations for a cone and for a wedge. The geometrical representations used are shown in Figures 20 and 21. The results obtained with different amounts of geometry detail in the longitudinal direction (ΔX/body length) are shown in Figures 22 and 23. These figures show that the program calculated values of C_{m_q} are within less than one percent of the exact analytical results when the shape is represented with longitudinal divisions of 10 percent of body length or less. It is evident from these results that the normal definition of a vehicle as used to give accurate static coefficients is also quite adequate for the dynamic derivatives.

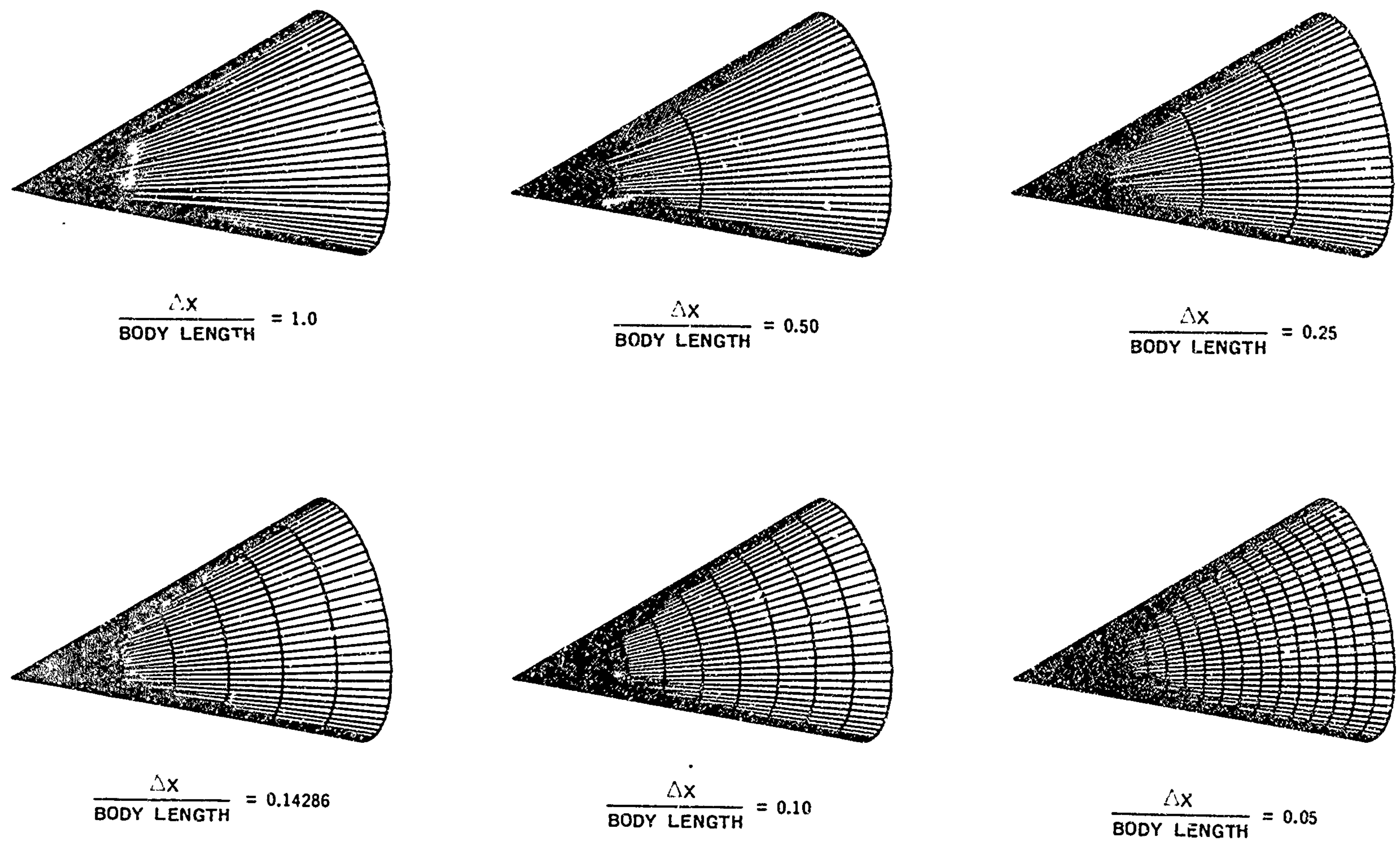


Figure 20. 20° Half Angle Cone. Graphical Representation of Various $\frac{\Delta X}{\text{Body Length}}$ Selections

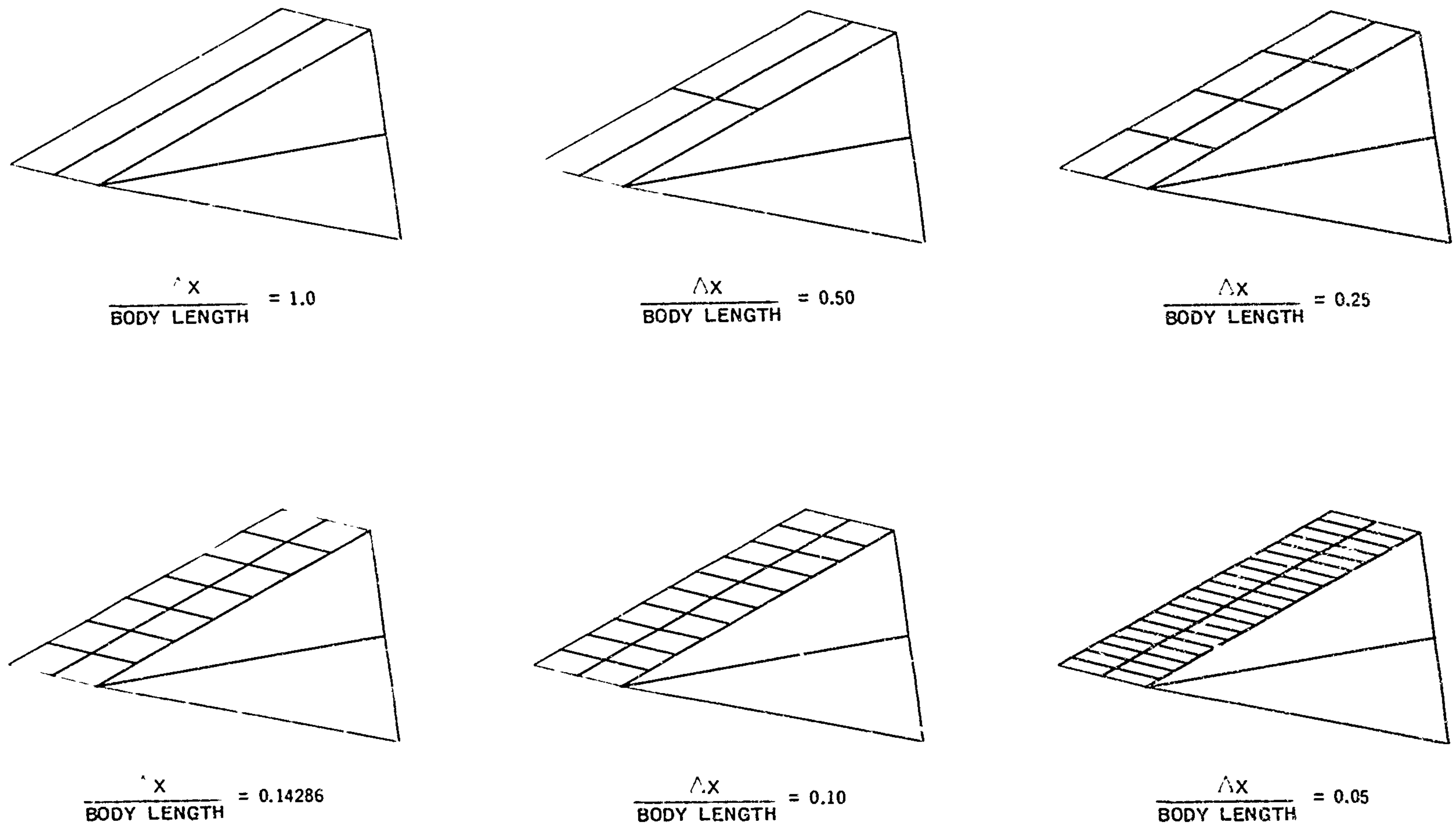


Figure 21. 20° Half Angle Wedge. Graphical Representation of Various $\frac{\Delta X}{\text{Body Length}}$ Selections

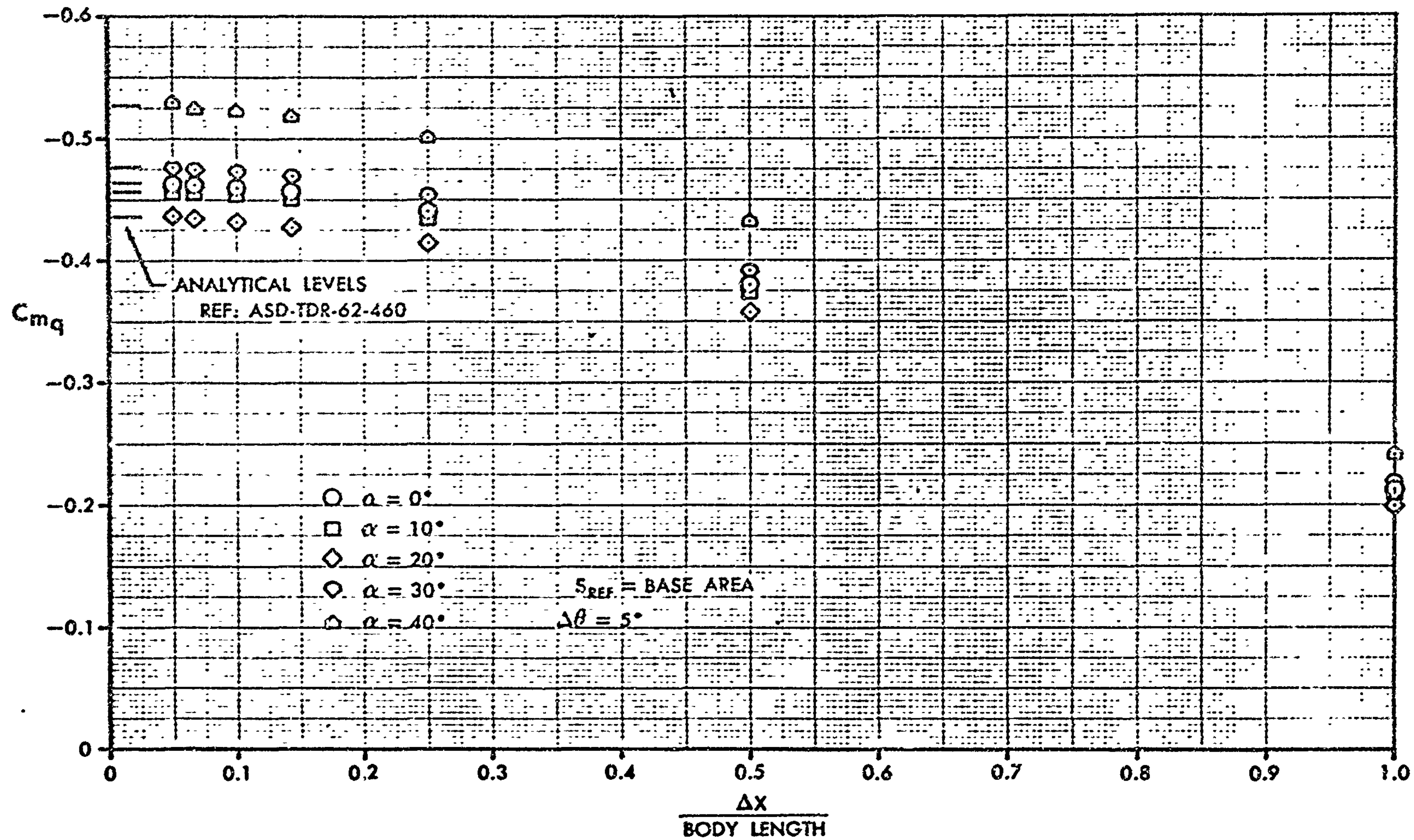


Figure 22. 20° Half Angle Cone. Effect of $\frac{\Delta X}{\text{Body Length}}$ on C_{m_q} Calculation

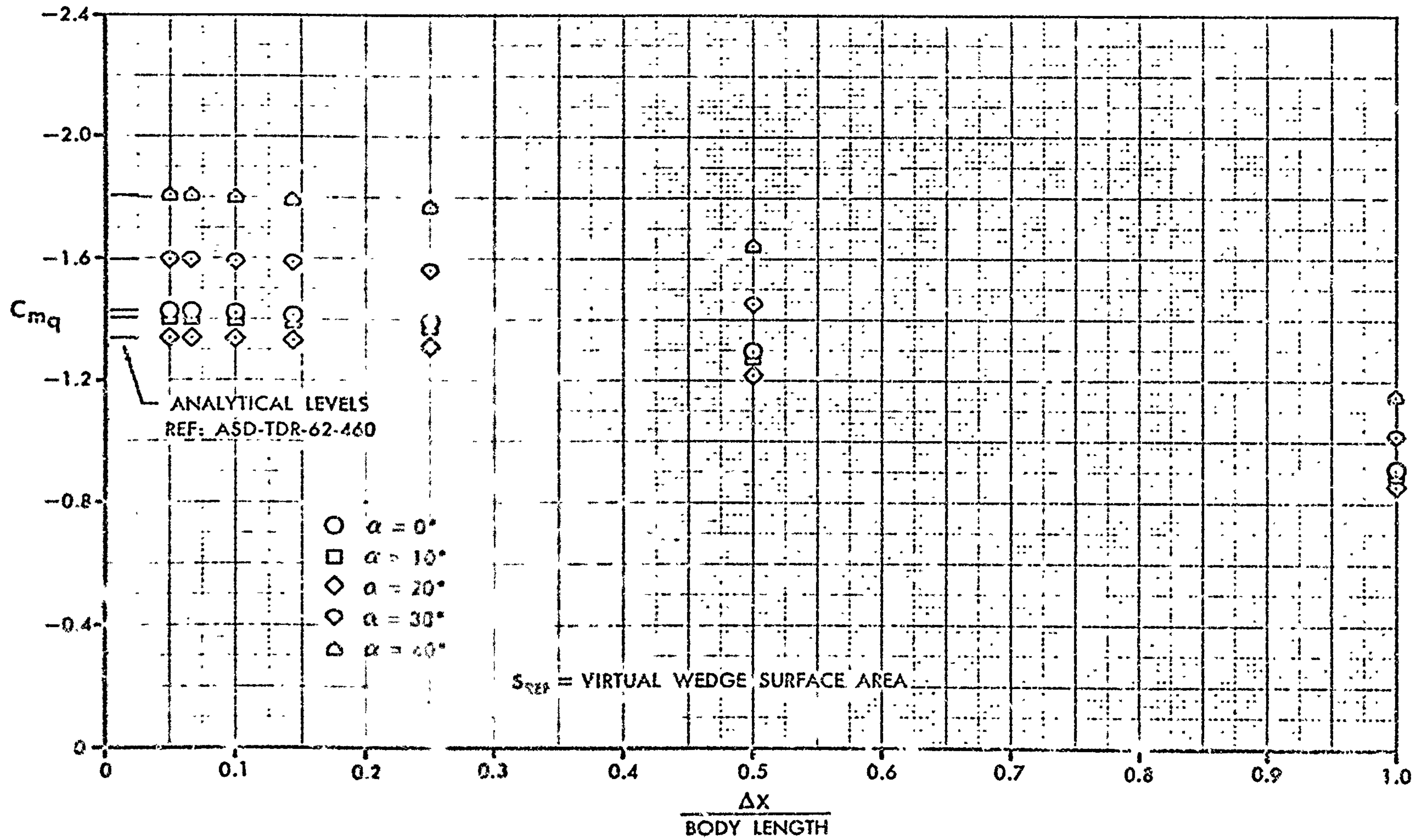


Figure 23. 20° Half Angle Wedge. Effect of $\frac{\Delta X}{\text{Body Length}}$ on C_{m_q} Calculation

The derivative $C_{m_{\dot{\alpha}}}$ is much more difficult to calculate accurately since it is caused by unsteady flow phenomena. Fortunately, this term is quite small for hypersonic vehicles in high-speed flight so approximate methods usually suffice. This stability derivative is the change in pitching moment with rate of change of angle of attack and occurs at constant pitch angle. It is frequently referred to as the "plunge" derivative since it represents the effect of a change in vertical acceleration. The common definition of this derivative is

$$C_{m_{\dot{\alpha}}} = \frac{\partial C_m}{\partial \left(\frac{\dot{\alpha} \bar{c}}{2V}\right)}$$

For a rigid vehicle this derivative arises from an aerodynamic time lag effect since the flow is not able to instantaneously adjust to changes in flight conditions. For low-speed airplanes the major component of this coefficient is caused by the fact that a change in the wing downwash field takes a finite length of time before it arrives at the tail. For tailless aircraft there still is a $C_{m_{\dot{\alpha}}}$ since the vehicle must accelerate the air mass in its path as it accelerates. This parameter is also subject to high-speed aeroelastic effects.

No general theory is currently available for calculating $C_{m_{\dot{\alpha}}}$. The parameter itself is usually difficult to obtain experimentally since it cannot be separated from the total damping coefficient measured $\left(C_{m_{\dot{\alpha}}} + C_{m_q} \right)$. References 60, 61, and 62 contain discussions of the most frequently used procedures for calculating $C_{m_{\dot{\alpha}}}$ for wing-body-tail configurations. Reference 63 discusses the use of the rheoelectric analogy for stability derivative determination.

In the analysis for the damping derivative $C_{m_{\dot{\alpha}}}$ it is convenient to divide the vehicle into separate components consisting of the body, wing, and tail surface. The wing contribution may be found from the relationship below.

$$\left(C_{m_{\dot{\alpha}}}\right)_w = \frac{S_w}{S}\left(K_{wB} + K_{Bw}\right)\left(\frac{\bar{c}_w}{\bar{c}}\right)^2\left(C'_{m_{\dot{\alpha}}}\right)_w$$

where

S_w = exposed wing area

S = reference area

K_{wB}	=	interference factor for effect of wing in presence of body (see Reference 64)
K_{Bw}	=	interference factor for effect of body in presence of wing
$\bar{c}_w$	=	mean aerodynamic chord of exposed wing
$\bar{c}$	=	reference chord for moment coefficients
$C'_{m_{\dot{\alpha}_w}}$	=	basic wing $\dot{\alpha}$ derivative as obtained from Reference 60.

The body contribution may be obtained by using the relatively simple results derived from slender-body theory. Reference 65 points out that although slender-body theory alone does not accurately predict the characteristics of nonslender configurations, the ratio of slender-body derivatives may be used to obtain reasonably accurate results as indicated below.

$$\left(C_{m_{\dot{\alpha}}}\right)_B = \left(C_{m_\alpha}\right)_B \left(\frac{C_{m_{\dot{\alpha}}}}{C_{m_\alpha}}\right)_{\text{slender body}}$$

where

$\left(C_{m_\alpha}\right)_B$	=	body pitching moment derivative as calculated by the arbitrary body program for the actual shape involved

$$\left(\frac{C_{m_{\dot{\alpha}}}}{C_{m_\alpha}}\right)_{\text{slender body}} = \frac{-4\,(L/\bar{c})^2\,\frac{S_B}{S}\,\frac{\text{Volume}}{S_B\,L}\left(\frac{x_o}{L} - \frac{x_c}{L}\right)}{2\,(L/\bar{c})\,\frac{S_B}{S}\left[\frac{\text{Volume}}{S_B\,L} - \left(1 - \frac{x_o}{L}\right)\right]}$$

x_o/L and x_c/L	=	center-of-gravity and area-centroid locations relative to the overall body length
S_B	=	body frontal area

The contributions of a horizontal tail to $C_{m_{\dot{\alpha}}}$ are given by the relationship

$$\left(C_{m_{\dot{\alpha}}}\right)_T = \frac{2QS_T}{S} \cos^2 \Gamma_T \left(C_{L_\alpha}\right)_T \left(K_{TB} + K_{BT}\right)\left(\frac{\bar{x}_T}{\bar{c}}\right)^2 \overline{\frac{\partial}{\partial Z}\left(\frac{\Omega_\alpha}{\alpha V}\right)}$$

where

Q = tail effectiveness ratio = $\dfrac{q_1 \left(C_{L_\alpha}\right)_1}{q_\infty \left(C_{L_\alpha}\right)_\infty}$

S_T = tail exposed area

Γ_T = tail dihedral angle

$\left(C_{L_\alpha}\right)_T$ = tail lift curve slope

K_{TB} and K_{BT} = tail lift interference factors

$\bar{x}_T$ = tail length

$\overline{\frac{\partial}{\partial Z}\left(\frac{\Omega_\alpha}{\alpha V}\right)}$ = average upwash induced by the wing at the tail location (becomes negligible in the hypersonic range)

The accuracy of the above methods for high angle of attack conditions or for very blunt lifting re-entry bodies is not known, although Reference 63 states that slender body theory is useful for such situations.

The use of the above relatively simple approach for the calculation of the $C_{m_{\dot{\alpha}}}$ derivatives would allow the determination of the $\dot{\beta}$ derivatives in a similar manner (see Reference 65).

$$\left(C_{Y_{\dot{\beta}}}\right)_B = \left(C_{Y_\beta}\right)_B \left(\frac{C_{Y_{\dot{\beta}}}}{C_{Y_\beta}}\right)_{\text{slender body}}$$

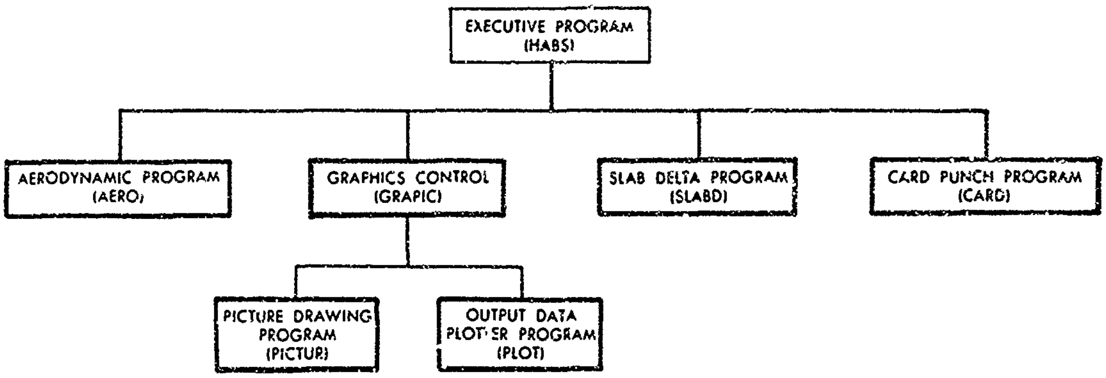
'ere

$$\left(\frac{C_{Y_{\dot{\beta}}}}{C_{Y_\beta}}\right)_{\text{slender body}} = \frac{4\,\frac{\text{Volume}}{Sb}}{-2\,\frac{S_B}{S}}$$

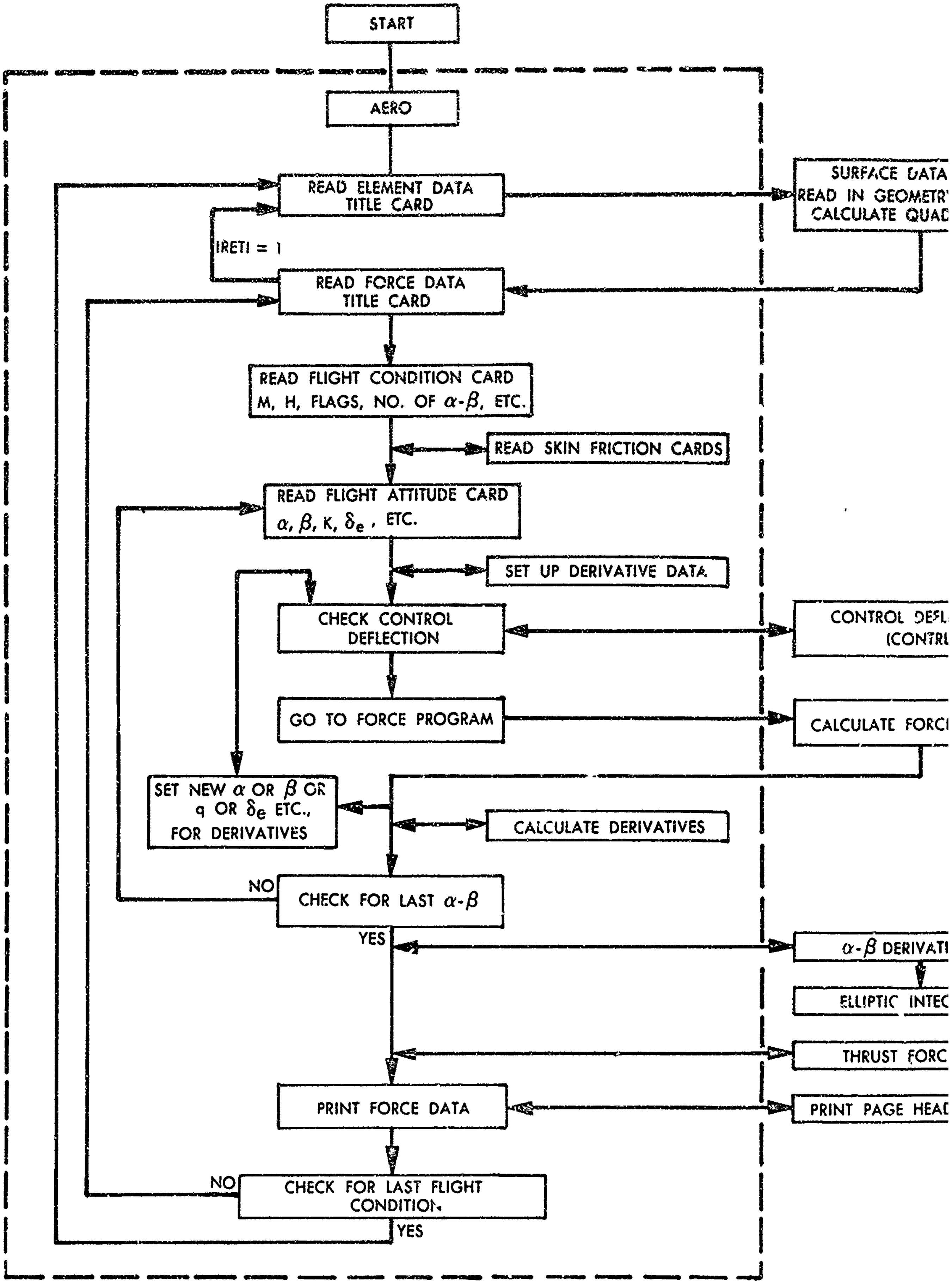
SECTION III

PROGRAM ORGANIZATION

The basic organization of the Mark III version of the Arbitrary-Body Program is shown in the diagram below. Each major part of the program could, with only minor modifications, operate as a completely separate computer program.



The most important component of this system is the Aerodynamic Program (AERO). The general features of the logic flow for this part of the system are shown on Figure 24. Each subroutine is described in more detail in Volume I of this report (pages 16 through 25). The graphics parts of the system are described on pages 25 and 26 of Volume I.



START
AERO
READ ELEMENT DATA TITLE CARD
SURFACE DATA
READ IN GEOMETR
CALCULATE QUAD
IRETI = 1
READ FORCE DATA TITLE CARD
READ FLIGHT CONDITION CARD
M, H, FLAGS, NO. OF α-β, ETC.
READ SKIN FRICTION CARDS
READ FLIGHT ATTITUDE CARD
α, β, K, δe, ETC.
SET UP DERIVATIVE DATA
CHECK CONTROL DEFLECTION
CONTROL DEFL
(CONTR
GO TO FORCE PROGRAM
CALCULATE FORC
SET NEW α OR β OR q OR δe ETC., FOR DERIVATIVES
CALCULATE DERIVATIVES
NO
CHECK FOR LAST α-β
YES
α-β DERIVAT
ELLIPTIC INTEC
THRUST FORC
PRINT FORCE DATA
PRINT PAGE HEAD
NO
CHECK FOR LAST FLIGHT CONDITION
YES

A

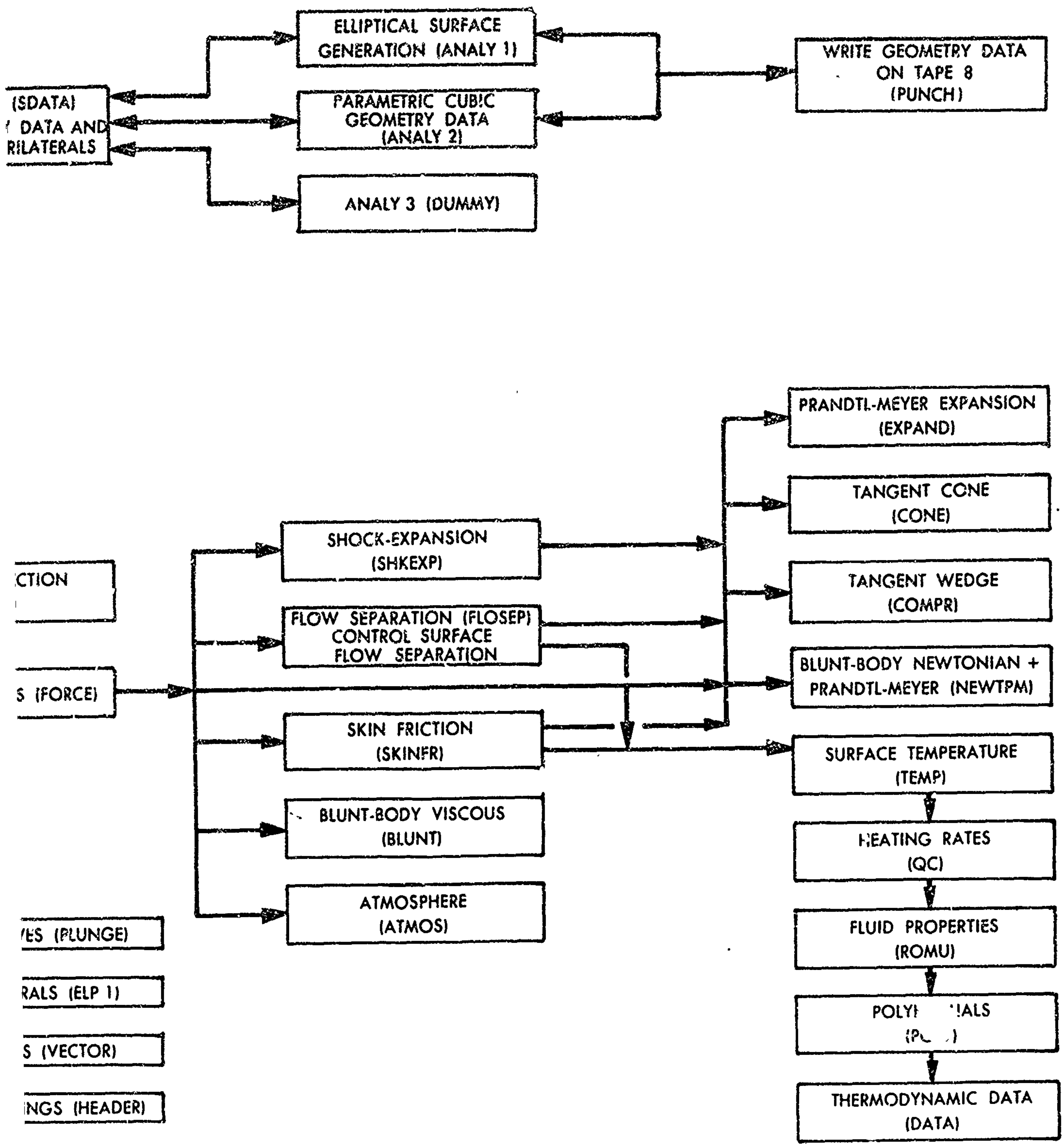


Figure 24. AERO Program Flow Chart

B

SECTION IV

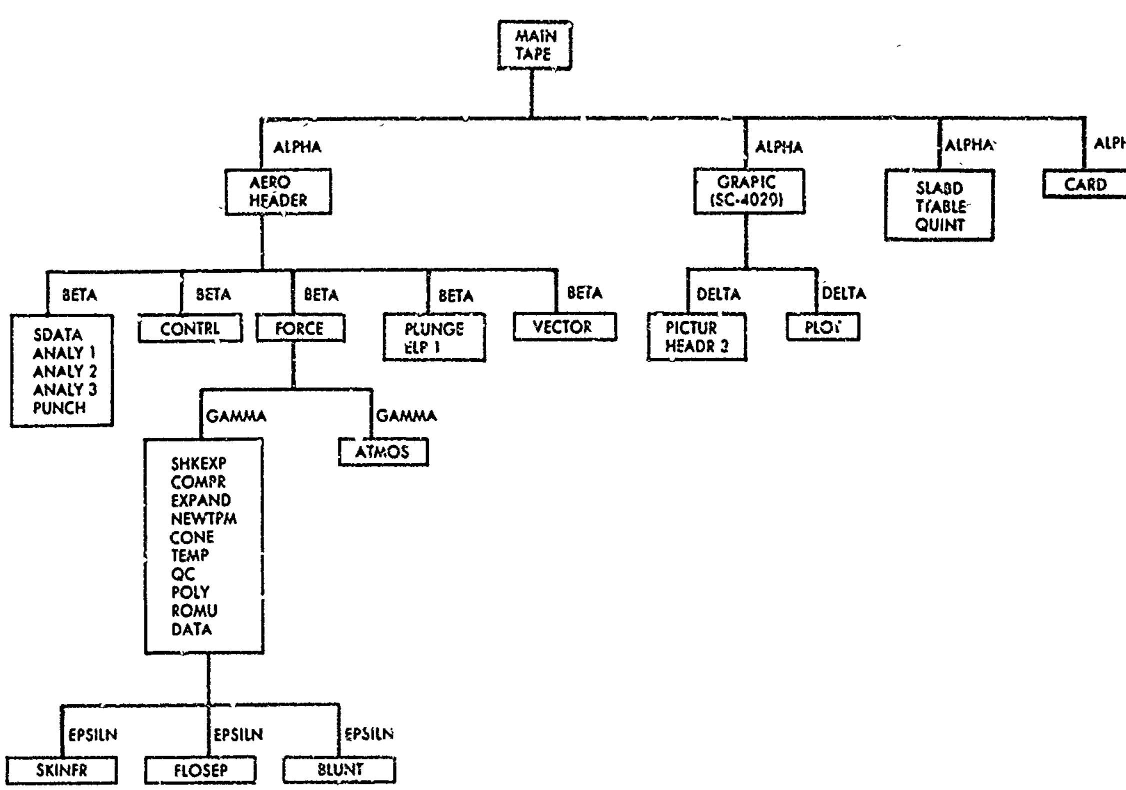
OPERATIONAL CONSIDERATIONS

The Mark III version of the Hypersonic Arbitrary-Body Aerodynamic Computer Program System is written entirely in FORTRAN. Models of the Mark III program are available for use on the IBM 360, the IBM 7094, and the UNIVAC 1108 computers. For the Mark III program the IBM 360 model is considered to be the base-line program. The differences between the programs are of a minor nature and modifications necessary for operation on other similar computers should be easy to accomplish. The program also makes use of the Douglas version of the SC-4020 software system package to generate a plotting tape. This tape is then processed off-line by a Stromberg Carlson SC-4020 Data Recording System. The graphics parts of the program may be converted by the user for operation with on-line graphics equipment such as the IBM 2250.

OVERLAY STRUCTURE

Because of the large size of this program it is necessary to use the overlay feature of FORTRAN. The overlay structure for the Mark III program is shown below. With this structure the program requires approximately 105,000 bytes of storage on the IBM 360 computer. The program will also operate on an IBM 7094 (32K) with the requirement that eleven tape drives be available (including the standard input Tape 5, output Tape 6, and the SC-4020 output tape).

OVERLAY STRUCTURE
MARK III PROGRAM



DECK SET-UP AND OPERATION

The overlay structure cards for the IBM 360 computer are given below.

```
ENTRY MAIN
INSERT MAIN=
OVERLAY ALPHA
INSERT AERO=, HEADER=
OVERLAY BETA
INSERT SDATA=, ANALY1=, ANALY2=, ANALY3=, PUNCH=
OVERLAY BETA
INSERT CONTRL=
OVERLAY BETA
INSERT FORCE=
INSERT IHCSERF
OVERLAY GAMMA
INSERT SHKEXP=, COMPR=, EXPAND=, NEWTPM=, TEMP=, QC=,
        POLY=, ROMU=, CONE=
OVERLAY EPSILN
INSERT FLOSEP=
OVERLAY EPSILN
INSERT SKINFR=
OVERLAY EPSILN
INSERT BLUNT=
OVERLAY GAMMA
INSERT ATMOS=
OVERLAY BETA
INSERT PLUNGE=, ELP1=
OVERLAY BETA
INSERT VECTOR=
OVERLAY ALPHA
INSERT GRAPHIC=
INSERT NXV=, XSCALV=, NHCAWAU, NHCAWAO, NHCAWAP  *
INSERT YMODV=, XMODV=, NHCAWBN  *
INSERT NHCAWFC, VXAXV=, APLOTV=, APRNTV=, BNBCDV=,  *
        BRITEV=, DOTLNV=     *
INSERT ERMRKV=, ERRLNV=, ERRNLV=, GRID1V=, LABLV=, LINEV=,  *
        LINRV=    *
INSERT NOFRV=, NONLNV=, PLOTV=, SETCIV=   *
INSERT SCSETV=, ARGQ=    *
OVERLAY DELTA
INSERT PICTUR=, HEADR2=
OVERLAY DELTA
INSERT PLOT=
OVERLAY ALPHA
INSERT SLABD=, TTABLE =, QINT=
OVERLAY ALPHA
INSERT CARD=
```

Note: Those cards marked with an asterisk (*) contain the SC-4020 routines. The asterisk is not actually punched on the cards but is just used in this report to indicate the SC-4020 insert cards.

The deck set-up for the IBM 7094 model of this program is outlined below.

$JOB	as required	
$SETUP	as required	
$RESTORE	required on Douglas system	
$*	as required to give on-line message to machine operator (any text in card columns 3-72)	
$EXECUTE	IBJOB	
$IBJOB		
HABS	Main program	
$ORIGIN	ALPHA	
AROA	Subroutine	AERO
AROB		HEADER
$ORIGIN	BETA	
AROC	Subroutine	SDATA
AROD		ANALY1
AROE		ANALY2
AROF		ANALY3
AROG		PUNCH
$ORIGIN	BETA	
AROH	Subroutine	CONTRL
$ORIGIN	BETA	
AROI	Subroutine	FORCE
$INCLUDE	FERF	
$ORIGIN	GAMMA	
AROJ	Subroutine	SHKEXP
AROM		COMPR
ARON		EXPAND
AROO		NEWTPM
AROP		CONE
AROR		TEMP
AROS		QC
AROT		POLY
AROU		ROMU
AROV		DATA

IBM 7094 Deck Set-Up (Continued)

$ORIGIN	EPSILN	
AROL	Subroutine	SKINFR
$ORIGIN	EPSILN	
AROK	Subroutine	FLOSEP
$ORIGIN	EPSILN	
AROQ	Subroutine	BLUNT
$ORIGIN	GAMMA	
AROW	Subroutine	ATMOS
$ORIGIN	BETA	
AROX	Subroutine	PLUNGE
AROY		ELP1
$ORIGIN	BETA	
AROZ	Subroutine	VECTOR
$ORIGIN	ALPHA	
GRPA	Subroutine	GRAPIC
$INCLUDE	All SC-4020 subroutines required by program except those that must be in the main link.	
$ORIGIN	DELTA	
GRPB	Subroutine	PICTUR
GRPC		HEADER2
$ORIGIN	DELTA	
GRPD	Subroutine	PLOT
$ORIGIN	ALPHA	
SLBA	Subroutine	SLABD
SLBB		TTABLE
SLBC		QINT
$ORIGIN	ALPHA	
CARD	Subroutine	CARD

$DATA
All program input data cards

End of File (7 and 8 punches in card column 1)

$IBSYS

TAPE ASSIGNMENTS

The operation of this program requires the availability of 11 logical units (besides the systems units). On the IBM 7094 these will be magnetic tape units. On the IBM 360 computer some or all of these units will be on disk storage. In either case, the logical units are referred to as "Tapes" throughout this discussion. The units required and their use in the program are listed below.

Unit	Mode	Program Usage
1	Binary	Storage of aerodynamic coefficients for summation.
3	Binary	Storage of control fore-surface geometry data.
4	Binary	Storage of excess of quadrilateral element data when number of elements is greater than ISIZE. Also used to store geometry data for control-surface flap.
5	BCD	Standard system input tape unit.
6	BCD	Standard system output tape unit.
-	Binary	Standard system punch tape unit.
8	BCD	Storage of geometry data in surface-element form.
9	Binary	Storage of aerodynamic data to be plotted by PLOT routine.
10	Binary	Storage of counter of aerodynamic data saved for plotting by PLOT routine.
11	Binary	Storage of control-surface geometry in the flap-deflected position.
16	Binary	SC-4020 output tape for the Douglas version of the SC-4020 system.

Tape Assignments (Continued)

For operation on the IBM 360 computer it is recommended that the following units be on magnetic tape: 3, 4, 8, and 11. These tapes may be defined as follows:

```
//GO.FT03F001  DD DSNAME=&FT03,UNIT=TAPEA,DISP=(NEW,DELETE),          X
//             DCB=(RECFM=V,BLKSIZE=800,DEN=2,TRTCH=C)
//GO.FT04F001  DD DSNAME=&FT04,UNIT=TAPEB,DISP=(NEW,DELETE),          X
//             DCB=(RECFM=V,BLKSIZE=800,DEN=2,TRTCH=C)
//GO.FT08F001  DD DSNAME=TFT08,UNIT=TAPEB,DISP=(NEW,KEEP),            X
//             DCB=(TRTCH=ET,RECFM=F,BLKSIZE=84,DEN=2)
//GO.FT11F001  DD DSNAME=&FT11,UNIT=TAPEB,DISP=(NEW,DELETE),          X
//             DCB=(RECFM=V,BLKSIZE=800,DEN=2,TRTCH=C)
```

SC-4020 SYSTEM

This program makes use of the Stromberg Carlson SC-4020 Data Recording System. The SC-4020 software system used by this program was originally developed by North American Aviation, Inc. The Douglas version of this system contains a number of revisions. Some of these revisions were made to facilitate the use of the overlay features of FORTRAN.

Because of the large core-storage requirements of the Hypersonic Arbitrary-Body System, it is important that all system subroutines be included in the proper overlay link. On the Douglas version of the SC-4020 system two subprograms must be overlay link 0. Other versions of the SC-4020 system may have similar restrictions. For the IBM 7094 model of the program these subroutines are PLO1 and SCOUTQ. Those subroutines required in the overlay structure for the IBM 360 model of the program are indicated by the asterisk on page 116. All other SC-4020 routines on the IBM 360 must be in the main link. The following cards may be used to include the appropriate SC-4020 subroutines in the proper part of the overlay structure for the IBM 7094 model of the program.

```
$INCLUDE     APLOTV,APRNTV,BUTTV,BNBCDV,BRITEV,CO4020,CH4020

$INCLUDE     DOTLNV,ER4020,ERMRKV,ERRLNV,ERRNLV,FR4020,GRID1V

$INCLUDE     HD4020,HOLDIV,HOLLV,ID4020,INCRV,INTBCD,LABLV

$INCLUDE     LE4020,LINEV,LINRV,NOFRV,NONLNV,PL4020,PLOTV

$INCLUDE     PRINTV,SCBSR,SETCIV,SETMIV,SMXYV,STOPTV,TF4020

$INCLUDE     TI4020,TPNUMV,WP4020,WR4020,XAXISV,XMODQ,XSCALQ
```

REFERENCES

1. Gentry, A. E., A Computer Program System for Calculating The Hypersonic Aerodynamic Characteristics of Arbitrary Three-Dimensional Shapes, Douglas Report No. DAC 56006 April 22, 1966.

2. Gentry, A. E., Hypersonic Arbitrary-Body Aerodynamic Computer Program (Mark II), Volume I — User's Manual and Volume II — Program Formulation and Listings, Douglas Report No. DAC-56080, March 1967.

3. Fowell, L. R., Flow Field Analysis for Lifting Re-Entry Configurations by the Method of Characteristics, IAS Paper No. 61-208-1902, June 1961.

4. Moretti, G., Sanlorenzo, E. A., Magnus, E. E., and Weilerstein, C., Supersonic Flow About General Three-Dimensional Bodies, Vol. III, Flow Field Analysis of Reentry Configurations by a General Three-Dimensional Method of Characteristics, ASD-TR-61-727, Vol. III, October 1962.

5. Batchelder, R. A., An Inverse Method for Inviscid Ideal Gas Flow Fields Behind Analytic Shock Shapes, Douglas Report No. SM 42588, July 1963.

6. Inouze, M., Rakich, J. V., and Lomax, H., A Description of Numerical Methods and Computer Programs for Two-Dimensional and Axisymmetric Supersonic Flow Over Blunt-Nosed and Flared Bodies, NASA TN D-2970, August 1965.

7. Creager, M. O., Effect of Leading-Edge Blunting on the Local Heat Transfer and Pressure Distributions Over Flat Plates in Supersonic Flow, NACA TN 4142, December 1957.

8. McLaughlin, E. J., Aerodynamic Analysis of Hypersonic Wings, ASD Technical Report 61-394, Volume I, October 1961.

9. Nonweiler, T., Delta Wings of Shapes Amenable to Exact Shock-Wave Theory, Journal of the Royal Aeronautical Society, Volume 67, January 1963.

10. Flower, J. W., Configurations for High Supersonic Speeds Derived from Simple Shock-Waves and Expansions, Journal of the Royal Aeronautical Society, Volume 67, May 1963.

11. Jones, J. G., A Method for Designing Lifting Configurations for High Supersonic Speeds Using the Flow Fields of Non-Lifting Cones, R. A. E. Report No. Aero 2674, A. R. C. 24846, 1963.

12. Kuchemann, D. and Sterne, L. H. G., Progress in Aeronautical Sciences, Volume 6, Pergamon Press, 1965.

13. Sims, J. L., Tables for Supersonic Flow Around Right Circular Cones at Zero Angle of Attack, NASA SP-3004, 1964.

14. Hankey, W. L., Jr., Optimization of Lifting Re-Entry Vehicles, ASD-TDR-62-1102, March 1963.

15. Malvestuto, F. S., Jr., et al, Study to Determine Aerodynamic Characteristics on Hypersonic Re-Entry Configurations, Part II, Analytical Phase, Volume 2, Design Charts, WADD TR 61-56, August 1962.

16. Clark, E. L., and Trimmer, L. L., Equations and Charts For The Evaluation of the Hypersonic Aerodynamic Characteristics of Lifting Configurations By the Newtonian Theory, AEDC-TDR-64-25, March 1964.

17. Jackson, C. M., Jr., A Semigraphical Method of Applying Impact Theory to an Arbitrary Body to Obtain the Hypersonic Aerodynamic Characteristics at Angle of Attack and Sideslip, NASA TN D-795, May 1961.

18. Dahlem, V., III, A Digital Computation Method for Determining the Hypersonic Aerodynamic Characteristics of Arbitrary Bodies, ASRMDF TM 62-61, September 1962.

19. Van Tassell, W., Free-Molecular and Newtonian Coefficients for Arbitrary Bodies, RAD-TM-63-63, August 1963.

20. Coons, S. A., Surfaces for Computer-Aided Design of Space Figures, Massachusetts Institute of Technology Preprint No. 299.

21. Gellert, G. O., Geometric Computing — Electronic Geometry for Semiautomated Design, Machine Design, March 18, and April 1, 1965.

22. Hess, J. L., and Smith, A. M. O., Calculation of Non-Lifting Potential Flow About Arbitrary Three-Dimensional Bodies, Douglas Report No. ES 40622, March 1962.

23. Love, E. S., Henderson, A., Jr., and Bertram, M. H., Some Aspects of the Air - Helium Simulation and Hypersonic Approximations, NASA TN D-49, October 1959.

24. Kaufman, L. G., II., Pressure Estimation Techniques for Hypersonic Flows Over Blunt Bodies, Journal of Astronautical Sciences, Volume X, No. 2, Summer 1963.

25. Ames Research Staff, Equations, Tables, and Charts for Compressible Flow, NACA TR 1135, 1953.

26. Korn, G. A., and Korn, T. M., Mathematical Handbook for Scientists and Engineers, McGraw-Hill, April 1961.

27. Liepmann, H. W., and Roshko, A., Elements of Gasdynamics, John Wiley and Sons, Inc., 1957.

28. Bertram, M. H. and Henderson, A., Jr., Recent Hypersonic Studies of Wings and Bodies, ARS Journal, Vol. 31, No. 8, August 1961.

29. Gregorek, G. M., Nark, T. C., and Lee, J. D., An Experimental Investigation of the Surface Pressure and the Laminar Boundary Layer on a Blunt Flat Plate in Hypersonic Flow, ASD-TDR-62-792, Volume I, March 1963.

30. Van Dyke, M. D., A Study of Hypersonic Small-Disturbance Theory, NACA Report 1194, 1954.

31. Shapiro, A. H., The Dynamics and Thermodynamics of Compressible Fluid Flow, The Ronald Press, 1953.

32. Hayes, W. D., and Probstein, R. F., Hypersonic Flow Theory, Academic Press, 1959.

[redacted]

34. Lukasiewicz, J., Hypersonic Flow-Blast Analogy, AEDC-TR-61-4, June 1961.

35. Jacobs, W. F., A Simplified Approximate Method for the Calculation of the Pressure Around Conical Bodies of Arbitrary Shape in Supersonic and Hypersonic Flow, JAS, December 1961, pp 987-988.

36. Sivells, J. C. and Payne, R. G., A Method of Calculating Turbulent-Boundary-Layer Growth at Hypersonic Mach Numbers, AEDC-TR-59-3, March 1959.

37. Monaghan, R. J., An Approximate Solution of the Compressible Laminar Boundary Layer on a Flat Plate, British R&M No. 2760, 1953.

38. Spalding, D. B. and Chi, S. W., The Drag of a Compressible Turbulent Boundary Layer on a Smooth Flat Plate With and Without Heat Transfer, J. Fluid Mech., Volume 18, Pt. 1, January 1964, pp 117-143.

3[illegible]

40. Lewis, C. H. and Burgess, E. C. III, Empirical Equations for the Thermodynamic Properties of Air and Nitrogen to 15,000°K, AEDC-TDR-63-138, July 1963.

41. Kaups, K. and Keltner, G., Laminar Compressible Boundary Layer on a Yawed Infinite Wing, Douglas Report No. LB 32706, March 1967.

42. Hilsenrath, J. and Klein, M., Tables of Thermodynamic Properties of Air in Chemical Equilibrium Including Second Virial Corrections from 1500° K to 15,000° K, AEDC-TR-65-58, March 1965.

43. Hansen, C. F., Approximations for the Thermodynamic and Transport Properties of High Temperature Air, NACA TN-4150, 1958.

44. Michel, R. and Kretzschmar, G., Determination Theorique D'une Enthalpie de Reference Pour la Couche Limite Laminaire de la Plaque Plane, Influence due Nombre de Prandtl, O.N.E.R.A. Report T.P. No. 41, 1963.

45. Eckert, E. R. G., Survey on Heat Transfer at High Speeds, WADC TR 54-70, April 1954.

46. Bromley, L. A. and Wilke, C. R., Viscoscity at Low Temperatures, University of California TR HE-150-157.

47. Sieron, T. R., and Martinez, C., Jr., Effects and Analysis of Mach Number and Reynolds Number on Laminar Skin Friction at Hypersonic Speeds, AFFDL-TR-65-5, April 1965.

48. Bertram, M. H., and Blackstock, T. A., Some Simple Solutions to the Problem of Predicting Bounary-Layer Self-Induced Pressures, NASA TN D-798, April 1961.

49. White, F. M., Jr., Hypersonic Laminar Viscous Interactions on Inclined Flat Plates, ARS Journal, May 1962.

50. Bertram, M. H., Hypersonic Laminar Viscous Interaction Effects on the Aerodynamics of Two-Dimensional Wedge and Triangular Planform Wings, NASA TN D-3523, August 1966.

51. Neal, L. Jr., Average Pressure and Skin Friction on Tapered Plates Hypersonic Flow, J. Spacecraft, Volume 3, No. 6, June 1966, pp 943-946.

52. Goldberg, L., The Structure of the Viscous Hypersonic Shock Layer, General Electric Report R65SD50, December 1965.

53. Goldberg, L., Forces and Moments on the Front Face of a Blunt Lifting Reentry Vehicle, General Electric Report R66SD21, April 1966.

54. Grabau, M., A Method of Forming Continuous Empirical Equations for the Thermodynamic Properties of Air From Ambient Temperatures to 15,000° K, With Applications, AEDC-TN-59-102, August 1959.

55. Griffith, B. J., Comparison of Data from the Gemini Flights and AEDC-VKF Wind Tunnels, AEDC-TR-66-178, October 1966.

56. Popinski, Z., and Ehrlich, C. F., Development Design Methods for Predicting Hypersonic Aerodynamic Control Characteristics, AFFDL-TR-66-85, September 1966.

57. Kaufman, L. G., II, Meckler, L., Hartofilis, S. A., and Weiss, D., An Investigation of Hypersonic Flow Separation and Control Characteristics, AFFDL-TR-174, January 1965.

58. Kaufman, L. G., II, et al, A Review of Hypersonic Flow Separation and Control Characteristics, ASD TDR 62-168, March 1962.

59. Erdos, J., and Pallone, A., Shock-Boundary Layer Interation and Flow Separation, Proceedings of the 1962 Heat Transfer and Fluid Mechanics Institute, Stanford University Press, June 1962.

60. Harmon, S. M., Stability Derivatives at Supersonic Speeds of Thin Rectangular Wings with Diagonal Ahead of Tip Mach Lines, NACA Report 925, 1949.

61. Henderson, A., Jr., Pitching-Moment Derivatives C_{m_q} and $C_{m_{\dot{\alpha}}}$ at Supersonic Speeds for a Slender-Delta-Wing and Slender-Body Combination and Approximate Solutions for Broad-Delta-Wing and Slender-Body Combinations, NACA TN 2553, December 1951.

62. Tobak, M., Reese, D. E., Jr., and Beam, B. H., Experimental Damping in Pitch of 45° Triangular Wings, NACA RM A50J26, December 1950.

63. Cahn, M. S. et al., Rapid Estimates of Stability Derivatives for Reentry Vehicles by Rheoelectric Analogy, AIAA Paper No. 66-58, January 1966.

64. Pitts, W. C., Meilsen, J. N., and Kaattari, G. E., Lift and Center of Pressure of Wing-Body-Tail Combinations at Subsonic Transonic, and Supersonic Speeds, NACA Report 1307, 1957.

65. Walker, H. J., and Wolowicz, C. H., Theoretical Stability Derivatives for the X-15 Research Airplane at Supersonic and Hypersonic Speeds Including a Comparison With Wind-Tunnel Results, NASA TM X-287, August 1960.

APPENDIX A

PROGRAM LISTINGS AND FLOW CHARTS

This part of the report contains the listings and flow charts for the Mark III Mod O version of the Hypersonic Arbitrary-Body Aerodynamic Computer Program System. Also included are the symbol lists defining the variables used in each subroutine. A combined list of all program variables is included in Appendix B.

The symbol lists are divided into five fields which are described as follows:

(i) The first field contains the symbol

(ii) The second field contains the letters I, L, or R, denoting integer, logical, or real variable respectively.

(iii) The third field contains the letters A, C, D, U, denoting argument list, common, dimensioned, or undimensioned, variable respectively. The hierarchy of the above letters is A, C, D, U.

(iv) The fourth field contains the definition of the symbol.

(v) The fifth field contains the name of the subroutine in which the symbol occurs.

INDEX TO SUBROUTINE LISTINGS AND FLOW CHARTS

1. MAIN PROGRAM (DECK HABS)

a. Algorithm

This routine, the main routine for the Mark III version of the Hypersonic Arbitrary-Body Aerodynamic Computer Program System, acts as the system monitor. It controls and initiates execution of the five principal components of this system. These are the Aerodynamic Force Analysis Program (Program AERO and its associated subroutines), the Graphics Program Options (the Picture Drawing Program and the Output Data Plotter Program), the Auxiliary Geometry Generation Program (Slab Delta Program), and the card punch routine.

The first action of the program is for this Executive routine to read in the System Control Card (Type 0). This card controls the selection of each program option and the order in which the options are to be used. Up to twenty different program phases may be used, and any given option may be used several times.

If an error has been detected in any of the program subroutines control will be returned to the Executive Program with the ERROR flag set to the appropriate value. If the error is of the non-fatal type the Executive Main Program will start reading in data cards until it finds a card with a Type number of 99. The program will then attempt to execute the next phase option.

b. Input/Output

System Control Card (Type 0)
Program Flow monitoring information

c. Error

An error condition occurs when an error is returned by a subroutine, or when the System Control Card contains an illegal value.

d. Subroutines Required

AERO, GRAPIC, SLABD, CARD

e. Argument List

Not applicable

f. Length

2170 bytes

DECK HABS

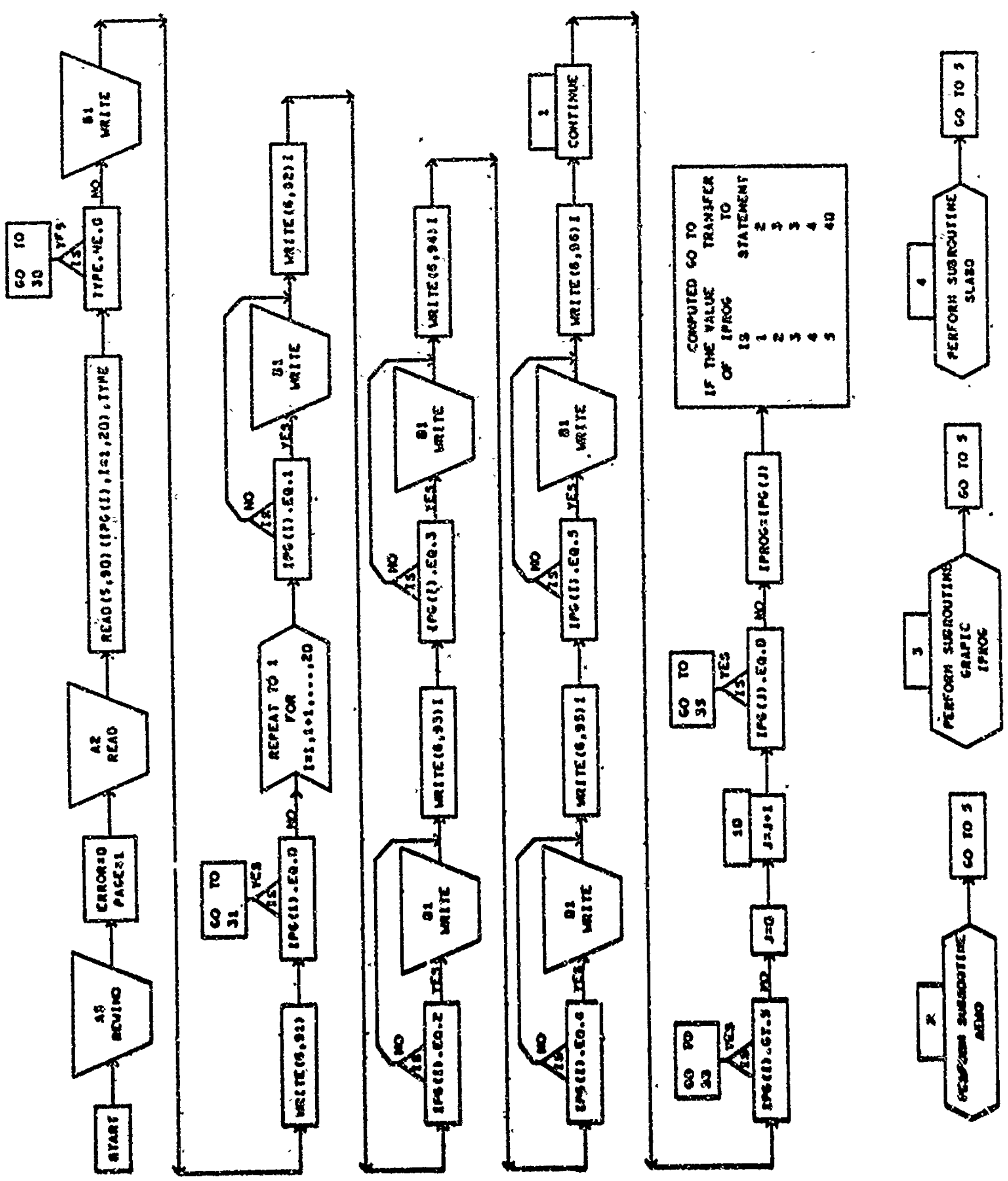
```
C************************************************************************
C*****                                                              *****
C**** HYPERSONIC ARBITRARY-BODY AERODYNAMIC COMPUTER PROGRAM SYSTEM ****
C*****                         IBM 360 MODEL                        *****
C************************************************************************
C
C     THIS IS THE  MARK III  MOD  0  VERSION OF THE HYPERSONIC
C     ARBITRARY BODY FORCE ANALYSIS PROGRAM.  THE BASIC PROGRAM
C     WAS DEVELOPED AT THE DOUGLAS AIRCRAFT COMPANY, AIRCRAFT
C     DIVISION UNDER SPONSORSHIP OF THE INDEPENDENT RESEARCH AND
C     DEVELOPMENT PROGRAM.  MODIFICATIONS TO THIS PROGRAM TO
C     INCORPORATE ADDITIONAL STABILITY AND CONTROL DERIVATIVES
C     AND PROGRAM DOCUMENTATION WHERE ACCOMPLISHED UNDER
C     AIR FORCE CONTRACT F 33615 67 C 1008 (SESSG, R.K. MILLS PROJECT
C     ENGINEER).  THIS VERSION WAS IDENTIFIED AS THE MARK II VERSION
C     AND WAS RELEASED IN MAY OF 1967 AND DOCUMENTED IN DOUGLAS
C     REPORT DAC 56080.  THE PRESENT MARK III VERSION IS AN
C     EXTENSIVELY MODIFIED VERSION OF THE EARLIER PROGRAM.  THESE
C     MODIFICATIONS WERE ACCOMPLISHED UNDER DOUGLAS IRAD AND ALSO
C     SUPPORTED IN PART BY AIR FORCE CONTRACT F 33615 67 C 1602
C     (FDMG, V. DAHLEM  PROJECT ENGINEER).
C
C     COMPLETE DOCUMENTATION OF THIS PROGRAM IS CONTAINED IN
C     DOUGLAS REPORT  DAC 61552
C         VOLUME 1  HYPERSONIC ARBITRARY-BODY AERODYNAMIC COMPUTER
C                   PROGRAM  MARK III VERSION - USERS MANUAL
C         VOLUME 2  HYPERSONIC ARBITRARY-BODY AERODYNAMIC COMPUTER
C                   PROGRAM  MARK III VERSION - PROGRAM FORMULATION
C                   AND LISTINGS
C
C************THIS PROGRAM WAS WRITTEN BY ARVEL E. GENTRY***************
C***************SENIOR ENGINEER  AERO-RESEARCH GROUP********************
C*****************REVISED SKIN FRICTION ROUTINES FOR*****************
C*****************THE MARK III PROGRAM BY D.N. SMYTH*****************
C************************************************************************
```

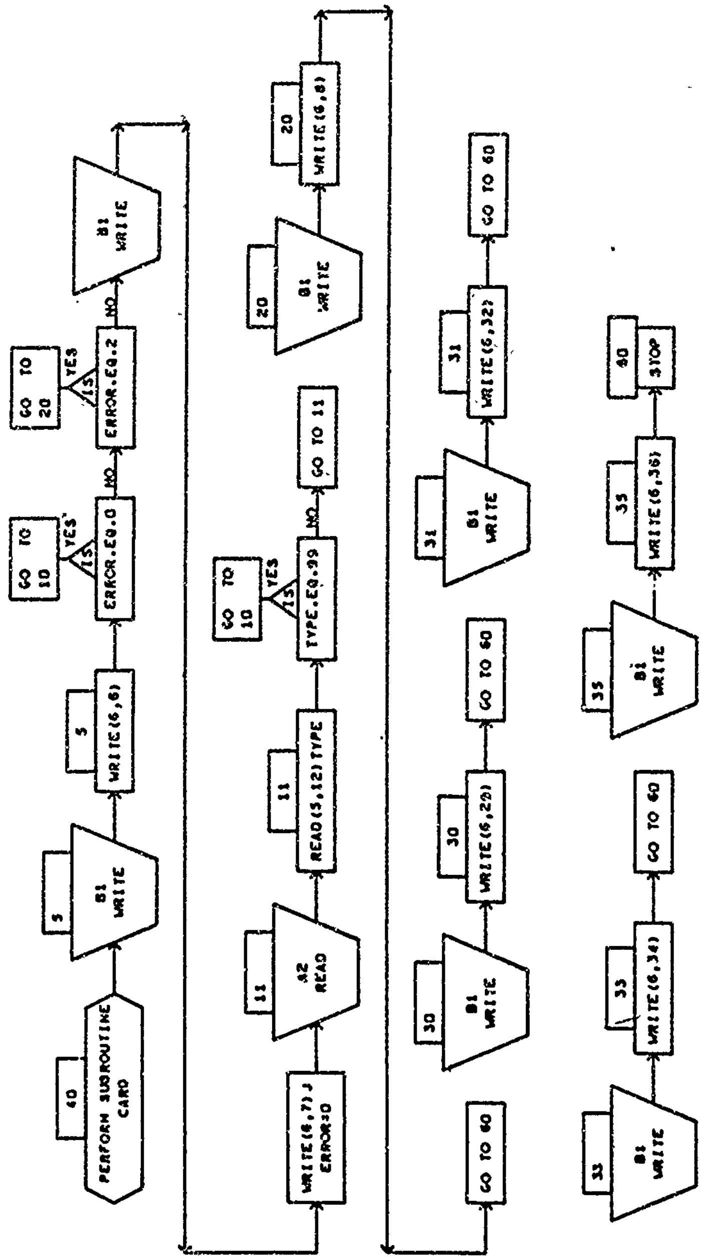
DECK HABS

```
C
C
C
C***** EXECUTIVE MAIN PROGRAM *****
C
      DIMENSION  IPG(20),TITLE(15)
      COMMON  CASE,TITLE,PAGE,ERROR
      INTEGER  TYPE,ERROR,CASE,PAGE
      REWIND 8
      ERROR = 0
      PAGE = 1
C
C ********************************************************************
C ***** READ SYSTEM CONTROL CARD TO SELECT PROGRAMS TO BE USED AND ORDER
C ***** OF USE. A MAXIMUM OF 20 OPTIONS MAY BE USED.  THIS VERSION OF
C ***** THE PROGRAM WILL ALLOW ONLY OPTIONS  1, 2, 3, 4, 5.
C*********************************************************************
C
      READ (5,90)  (IPG(I),I=1,20),TYPE
   90 FORMAT (20(2XI1),10X,I2)
      IF (TYPE .NE. 0) GO TO 30
      WRITE (6,91)
   91 FORMAT (1H1,////,/1H ,39H************* HYPERSONIC ARBITRARY-BODY
     1 50H AERODYNAMIC COMPUTER PROGRAM SYSTEM ************* ,///,1H ,
     2 10X,46HPROGRAM OPTIONS ARE IN THE FOLLOWING ORDER....   ,/1H  )
      IF (IPG(1) .EQ. 0) GO TO 31
C
C
      DO 1 I=1,20
      IF (IPG(I) .EQ. 1) WRITE (6,92) I
   92 FORMAT (1H0,15X,I2,40H  AERODYNAMIC FORCE PROGRAM    (OPTION 1) )
      IF (IPG(I) .EQ. 2) WRITE (6,93) I
   93 FORMAT (1H0,15X,I2,40H  PICTURE DRAWING PROGRAM      (OPTION 2) )
      IF (IPG(I) .EQ. 3) WRITE (6,94) I
   94 FORMAT (1H0,15X,I2,40H  OUTPUT DATA PLOTTER PROGRAM (OPTION 3) )
      IF (IPG(I) .EQ. 4) WRITE (6,95) I
```

```
DECK HABS

   95  FORMAT (1HO,15X,I2,40H  SLAB DELTA GEOMETRY PROGRAM (OPTION 4) )    HABS  0720
       IF (IPG(I) .EQ. 5) WRITE (6,96) I                                   HABS  0730
   96  FORMAT (1HO,15X,I2,41H  GEOMETRY DATA PUNCH PROGRAM (OPTION 5)  )    HABS  0740
C                                                                          HABS  0750
    1  IF (IPG(I) .GT. 5) GO TO 33                                         HABS  0760
       J = 0                                                               HABS  0770
C                                                                          HABS  0780
   10  J = J + 1                                                           HABS  0790
       IF (IPG(J) .EQ. 0) GO TO 35                                         HABS  0800
C                                                                          HABS  0810
       IPROG = IPG(J)                                                      HABS  0820
       GO TO (2,3,3,4,40), IPROG                                           HABS  0830
    2  CALL AERO                                                           HABS  0840
       GO TO 5                                                             HABS  0850
C                                                                          HABS  0860
    3  CALL GRAPIC (IPROG)                                                 HABS  0870
       GO TO 5                                                             HABS  0880
C                                                                          HABS  0890
    4  CALL SLABD                                                          HABS  0900
       GO TO 5                                                             HABS  0910
C                                                                          HABS  0920
   40  CALL CARD                                                           HABS  0930
C                                                                          HABS  0940
    5  WRITE (6,6)                                                         HABS  0950
    6  FORMAT (1H1,//////,1HO,40H******** MAIN PROGRAM NOW HAS CONTROL OF  HABS  0960
      1 16H SYSTEM ******** )                                              HABS  0970
       IF (ERROR .EQ. 0) GO TO 10                                          HABS  0980
       IF (ERROR .EQ. 2) GO TO 20                                          HABS  0990
       WRITE (6,7) J                                                       HABS  1000
    7  FORMAT (1HO,15X,35HA NON-FATAL ERROR OCCURRED IN PHASE ,I3,/15X,    HABS  1010
      1 57H   PROGRAM WILL ATTEMPT TO CONTINUE BY SEARCHING FOR NEXT       HABS  1020
      2 17H TYPE = 99  CARD.)                                              HABS  1030
       ERROR = 0                                                           HABS  1040
   11  READ (5,12) TYPE                                                    HABS  1050
   12  FORMAT (70X,I2)                                                     HABS  1060
       IF (TYPE .EQ. 99) GO TO 10                                          HABS  1070
```

```
DECK HABS

      GO TO 11                                                          HABS
C                                                                       HABS
   20 WRITE (6,8)                                                       HABS
    8 FORMAT (1H ,//,1H ,40H******** FATAL ERROR *** PROGRAM STOPPED )  HABS
      GO TO 60                                                          HABS
C                                                                       HABS
   30 WRITE (6,29)                                                      HABS
   29 FORMAT (1H0,50H ***** MASTER CONTROL CARD HAD TYPE NOT = 0 ******)HABS
      GO TO 60                                                          HABS
C                                                                       HABS
C                                                                       HABS
   31 WRITE (6,32)                                                      HABS
   32 FORMAT (1H ,//,1H ,38H***** FIRST PHASE OPTION IS ZERO *****      HABS
     1 21H**** FATAL ERROR **** )                                       HABS
      GO TO 60                                                          HABS
C                                                                       HABS
   33 WRITE (6,34)                                                      HABS
   34 FORMAT (1H ,//,1H ,42H***** PROGRAM OPTION IS GREATER THAN 5 ***  HABS
     1 22H***** FATAL ERROR ****  )                                     HABS
      GO TO 60                                                          HABS
C                                                                       HABS
   35 WRITE (6,36)                                                      HABS
   36 FORMAT (1H ,//,1H ,44H***** PROGRAM HAS REACHED NORMAL TERMINATION HABS
     1  7H ******  )                                                    HABS
   60 STOP                                                              HABS
      END                                                               HABS
```


START
A5 REWIND
ERROR=0 PAGE=1
A2 READ
READ(5,90)(IPG(I),I=1,20),ITYPE
ITYPE.NE.0
IS
YES
GO TO 30
NO
B1 WRITE
WRITE(6,91)
IPG(1).EQ.0
GO TO 31
REPEAT TO 1 FOR I=1,I+1,...,20
IPG(I).EQ.1
B1 WRITE
WRITE(6,92)I
IPG(I).EQ.2
WRITE(6,93)I
IPG(I).EQ.3
WRITE(6,94)I
IPG(I).EQ.4
WRITE(6,95)I
IPG(I).EQ.5
WRITE(6,96)I
1 CONTINUE
IPG(J).GT.5
GO TO 33
J=0
10 J=J+1
IPG(J).EQ.0
GO TO 35
IPROG=IPG(J)
COMPUTED GO TO
IF THE VALUE OF IPROG IS 1 2 3 4 5
TRANSFER TO STATEMENT 2 3 3 4 40
2 PERFORM SUBROUTINE AERO
GO TO 5
3 PERFORM SUBROUTINE GRAPIC IPROG
GO TO 5
4 PERFORM SUBROUTINE SLAB0
GO TO 5



40
PERFORM SUBROUTINE CARD
5
B1 WRITE
5
WRITE(6,6)
IS
ERROR.EQ.0
YES
GO TO 10
NO
IS
ERROR.EQ.2
YES
GO TO 20
NO
B1 WRITE
WRITE(6,7)
ERROR=0
11
A2 READ
11
READ(5,12) TYPE
IS
TYPE.EQ.99
YES
GO TO 10
NO
GO TO 11
20
B1 WRITE
20
WRITE(6,8)
GO TO 60
30
B1 WRITE
30
WRITE(6,29)
GO TO 60
31
B1 WRITE
31
WRITE(6,32)
GO TO 60
33
B1 WRITE
33
WRITE(6,34)
GO TO 60
35
B1 WRITE
35
WRITE(6,36)
40
STOP

```
SYMBOLS USED IN SUBROUTINE MAIN

CASE     I  C  PROBLEM CASE NUMBER                                          MAIN
ERROR    I  C  ERROR FLAG (=0 NO ERRUR, =1 NON-FATAL, =2 FATAL)             MAIN
I        I  U  DO-LOOP INDEX                                                MAIN
IPG      I  D  PROGRAM OPTION FLAG ARRAY                                    MAIN
IPROG    I  U  ACTIVE PROGRAM OPTION                                        MAIN
J        I  U  INDEX ON PROGRAM OPTION                                      MAIN
PAGE     I  C  PAGE NUMBER                                                  MAIN
TITLE    R  C  PROBLEM TITLE                                                MAIN
TYPE     I  U  CARD TYPE                                                    MAIN
```

2. SUBROUTINE AERO (DECK AROA)

This routine controls the flow of calculations within the Aerodynamic Force Calculation Option.

a. Algorithm

The first operation of this routine is to rewind the necessary program tape units. It then reads the Element Data Title Card (Type 1) and calls the Surface Data routine for the geometry calculations. Upon return from Surface Data (SDATA) the Force Data Title Card (Type 8) is read and if IRET1 is zero the Force Data Title Card will be read next. This is then followed by the Flight Condition Card, the Center of Gravity Data Card, and if required, the Skin Friction Data Cards and Coefficient Increment Data Cards. The program then starts the cycle to calculate force data at each flight attitude. This is initiated by the reading of the Flight Attitude Data Card. The necessary data arrays are initialized, derivative data established, and the Force calculation routine called.

Upon return from the FORCE routine the derivative cycle data is set and control returned to the FORCE routine if required. If the control surface is to be deflected the CONTROL routine will be called before the FORCE routine is called. If required the Plunge Derivative and Thrust Vector routines will also be called.

b. Input/Output

Element Data Title Card (Type 1), Force Data Title Card (Type 8), Flight Condition Card (Type 9), Center of Gravity Data Card (Type 10), Skin Friction Data Cards (Type 11), Flight Attitude Data Cards (Type 12), and Coefficient Increment Data Cards (Type 13). Output data consists of the vehicle summary force data.

c. Error

An error condition occurs when input card Type number is in error.

d. Subroutines Required

SDATA, CONTRL, FORCE, PLUNGE, VECTOR, HEADER

e. Argument List

None

f. Length

15396 bytes

DECK AROA

```
      SUBROUTINE AERO                                                  AROA 0010
C                                                                      AROA 0020
      DIMENSION   TITLE(15),ALP(20),BET(20),CCA(20),CCY(20),CCN(20),   AROA 0030
     1 CCLL(20),CCLM(20),CCLN(20),CCL(20),CCD(20),CLOD(20),CPS(20),    AROA 0040
     2 CF(20),CARD(20),QQINFS(20),IS(10,9),SURF(10,8),FS(8),BS(8),     AROA 0050
     3 CCDS(20),CCLS(20),CCAS(20),CCYS(20),CCNS(20),CCLMS(20),         AROA 0060
     4 CCLLS(20),CCLNS(20),CFS(20),DCAA(20),DCLA(20),DCNA(20),DCMA(20),AROA 0070
     5 DCMQ(20),DCAQ(20),DCNQ(20),DCYB(20),DCNB(20),DCLLB(20),DCYR(20),AROA 0080
     6 DCLNR(20),DCLLR(20),DCAAS(20),DCLAS(20),DCNAS(20),DCMAS(20),    AROA 0090
     7 DCMQS(20),DCAQS(20),DCNQS(20),DCYBS(20),DCNBS(20),DCLLBS(20),   AROA 0100
     8 DCYRS(20),DCLNRS(20),DCLLRS(20),IMP(20),ISH(20),IMPI(20),       AROA 0110
     9 ISHI(20),ETACS(20),ENPMS(20),QRPS(20),DELTES(20),IDERS(20),     AROA 0120
     A IPRINS(20),IPRTS(20),DCLD(20),DCMD(20),DCLLD(20),DCYD(20),      AROA 0130
     B DCLND(20),DCND(20),DCLDS(20),DCMDS(20),DCLLDS(20),              AROA 0140
     C DCYDS(20),DCLNDS(20),DCNDS(20),HMLS(20),HMRS(20),HML(20),       AROA 0150
     D HMR(20),DCMADT(20),DCYBDT(20),DCMADS(20),DCYBDS(20)             AROA 0160
C                                                                      AROA 0170
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),            AROA 0180
     1 YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)         AROA 0190
C                                                                      AROA 0200
C                                                                      AROA 0210
      COMMON   CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2, AROA 0220
     1 AREA2,IN,IM,L,LS,FS,BS,ALP,BET,CCA,CCY,CCN,CCLL,CCLM,CCLN,CCL,  AROA 0230
     2 CCD,CLOD,CF,CPS,QQINFS,IS,SURF                                  AROA 0240
      REAL   NX2,NY2,NZ2                                               AROA 0250
C                                                                      AROA 0260
      REAL   MACH,MAC                                                  AROA 0270
C                                                                      AROA 0280
      DATA   CCDS,CCLS,CCAS,CCYS,CCNS,CCLMS,                           AROA 0290
     1  CCLLS,CCLNS,CFS/180*0.0/                                       AROA 0300
      DATA   DCAAS,DCLAS,DCNAS,DCMAS,DCMQS,DCAQS,                      AROA 0310
     1   DCNQS,DCYBS,DCNBS,DCLLBS,DCYRS,DCLNRS,                        AROA 0320
     2   DCLLRS,DCLDS,DCMDS,DCLLDS,DCYDS,DCLNDS,                       AROA 0330
     3   DCNDS,HMLS,HMRS,DCMADS,DCYBDS/460*0.0/                        AROA 0340
C                                                                      AROA 0350
```

```
DECK ARDA

      DATA IPL9,IPL10/43,42/
C
      INTEGER  PAGE,CASE,SYMFCT,ERROR,TYPE,PRINT,PRINTS
C
      REWIND 1
      REWIND 9
      REWIND 10
      NSAVE = 0
      ISIZ =  300
C
C  SET INITIAL CONSTANTS FOR START OF CASE
   1  ERROR = 0
      REWIND 3
      REWIND 4
      REWIND 11
      DELTAS = 999.0
C
C  READ IN TITLE CARD (CARD COLUMN 1 THROUGH 60)
      READ (5,100) (TITLE(I),I=1,15),ISUM,IREW8,IPS,IABDOT,IVECT,CASE,
     1     TYPE
 100  FORMAT(14A4,A3,2I1,1X3I1,I3,2XI2)
      IF (TYPE .EQ. 99) GO TO 700
      IF (TYPE.NE.1) GO TO 800
      IF (IREW8 .NE. 0) REWIND 8
C
C  CHECK IF PREVIOUSLY CALCULATED DATA IS TO BE SUMMED (IF ISUM= 2 OR 3)
      IF (ISUM .GT. 1) GO TO 15
C
C  GO TO SURFACE DATA ROUTINE
      CALL SDATA (PRINTS,SYMFCT,ISIZ,IORIEN,IGTYPE)
C
C  CHECK CARD TYPE ERROR
      IF (ERROR.EQ.1) GO TO 800
      IF (ERROR .EQ. 3) GO TO 806
C
C  READ IN TITLE CARD (CARD COLUMN 1 THROUGH 60)
```

AERO

DECK AROA

```
   11 READ (5,110) (TITLE(I),I=1,15),IRFT1,CASE,TYPE
  110 FORMAT(14A4,1A3,I1,5XI3,2XI2)
      IF (TYPE.NE.8) GO TO 800
      IF (IRET1 .NE. 0) GO TO 1
C
C  READ IN VEHICLE CONDITION DATA CARDS (2 CARDS)
      READ (5,101) MACH,ALT,SREF,PSTAG,TSTAG,IPS,ITYP13,IABDOT,IVECT,
     1     LAST,NAB,NOAB,NS,TYPE
  101 FORMAT (3F10.0,1X,F5.0,F7.0,1XI1,2XI1,4X,2I1,5XI1,I2,I1,I2,5XI2)
      IF (MACH .GT. 1.0) GO TO 702
      WRITE (6,701)
  701 FORMAT (1H0,47H****INPUT MACH NUMBER MUST BE GREATER THAN  1.0   )
      ERROR = 1
      GO TO 806
  702 IFIRST = 0
      IF (TYPE.NE.9) GO TO 800
      READ (5,102) XCG,YCG,ZCG,SPAN,MAC,TYPE
  102 FORMAT (5F10.0,20XI2  )
      IF (TYPE .NE. 10) GO TO 800
      NABCT = 1
      NABS = NAB
      NPRT = 28
      IDFLGA = 0
      IDFLGB = 0
      IDFLGC = 0
      IDFLGD = 0
      IDFLGE = 0
      IDFLGF = 0
C
C  CHECK ON NUMBER OF ALPHA-BETA CONDITIONS (MUST BE LESS THAN 21)
      IF (NAB.LT.21) GO TO 7
      WRITE (6,108)
  108 FORMAT (1H ,47H***** NUMBER OF ALPHA-BETA CONDITIONS CANNOT BE
     1 59H GREATER THAN 20. JOB WILL BE ATTEMPTED WITH NAB = 20 ***** )
      NAB = 20
C
```

```
DECK AROA

C   READ SKIN FRICTION SURFACE DATA IF REQUIRED
   7  IF (NS .EQ. 0) GO TO 2
      DO 8 I=1,NS
      READ (5,103) (IS(I,J), J=1,9), (SURF(I,J), J=1,8), TYPE
 103  FORMAT (I2,8I1,2F9.0,3F6.0,2F6.0,F4.0,8XI2 )
      IF (TYPE .NE. 11) GO TO 800
   8  CONTINUE
C
C   START OF CYCLE TO CALCULATE DATA AT EACH ALPHA-BETA COMBINATION
C   READ ALPHA-BETA-CPSTAG-SKIN DATA CARD
   2  IF (NOAB .EQ. 1) GO TO 34
      READ (5,104) ALPHA,BETA,CPSTAG,QRP,ETAC,ENPM,QQINF,IMPACT,ISHAD,
     1  IDERIV,PRINT,IPRINT,NW,IMPACI,ISHADI,DELTAE,RETRAN,TWALL,TYPE
 104  FORMAT (3F6.0,F7.0,3F5.0,2I2,4I1,2I2,F5.0,2F4.0,5XI2)
C   CHECK CARD TYPE ERROR
      IF (TYPE .NE. 12) GO TO 802
      RETRAN = RETRAN * 1.0E6
      GO TO 35
C   SET FLIGHT CONDITION DATA TO PREVIOUS INPUT DATA
  34  ALPHA = ALP(NABCT)
      BETA  = BET(NABCT)
      CPSTAG= CPS(NABCT)
      QRP   = QRPS(NABCT)
      ETAC  = ETACS(NABCT)
      ENPM  = ENPMS(NABCT)
      QQINF = QQINFS(NABCT)
      IMPACT= IMP(NABCT)
      ISHAD = ISH(NABCT)
      IDERIV= IDERS(NABCT)
      PRINT = IPRINS(NABCT)
      IPRINT= IPRTS(NABCT)
      IMPACI= IMPI(NABCT)
      ISHADI= ISHI(NABCT)
      DELTAE= DELTES(NABCT)
      IF (NAB .EQ. 0) NAB = NABS
C
```

DECK AROA

```
   35  IMP(NABCT) = IMPACT                                        AROA  1440
       ISH(NABCT) = ISHAD                                         AROA  1450
       ETACS(NABCT) = ETAC                                        AROA  1460
       ENPMS(NABCT) = ENPM                                        AROA  1470
       QRPS(NABCT) = QRP                                          AROA  1480
       IDERS(NABCT) = IDERIV                                      AROA  1490
       IPRINS(NABCT) = PRINT                                      AROA  1500
       IPRTS(NABCT) = IPRINT                                      AROA  1510
C   SET ALL DERIVATIVES TO ZERO                                   AROA  1520
       DCAA(NABCT)  = 0.0                                         AROA  1530
       DCLA(NABCT)  = 0.0                                         AROA  1540
       DCNA(NABCT)  = 0.0                                         AROA  1550
       DCMA(NABCT)  = 0.0                                         AROA  1560
       DCMQ(NABCT)  = 0.0                                         AROA  1570
       DCAQ(NABCT)  = 0.0                                         AROA  1580
       DCNQ(NABCT)  = 0.0                                         AROA  1590
       DCYB(NABCT)  = 0.0                                         AROA  1600
       DCNB(NABCT)  = 0.0                                         AROA  1610
       DCLLB(NABCT) = 0.0                                         AROA  1620
       DCYR(NABCT)  = 0.0                                         AROA  1630
       DCLNR(NABCT) = 0.0                                         AROA  1640
       DCLLR(NABCT) = 0.0                                         AROA  1650
       DCLD(NABCT)  = 0.0                                         AROA  1660
       DCMD(NABCT)  = 0.0                                         AROA  1670
       DCLLD(NABCT) = 0.0                                         AROA  1680
       DCYD(NABCT)  = 0.0                                         AROA  1690
       DCLND(NABCT) = 0.0                                         AROA  1700
       DCND(NABCT)  = 0.0                                         AROA  1710
       DCMADT(NABCT)= 0.0                                         AROA  1720
       DCYBDT(NABCT)= 0.0                                         AROA  1730
C                                                                 AROA  1740
C   CHECK IF DERIVATIVES ARE TO BF CALCULATED                     AROA  1750
       IF (IDERIV .EQ. 0) GO TO 5                                 AROA  1760
C   SET UP STARTING DERIVATIVE DATA                               AROA  1770
       IDSTAT = 1                                                 AROA  1780
       ALPHAS = ALPHA                                             AROA  1790
```

DECK AROA

```
      BETAS   = BETA
      DELTS   = DELTAE
      QRPSS   = QRP
      IF (IDERIV.EQ.1 .OR. IDERIV.EQ.5) ALPHA = ALPHA + 1.0
      IF (IDERIV.EQ.2 .OR. IDERIV.EQ.6) BETA  = BETA  + 1.0
      IF (IDERIV .EQ. 4) DELTAE= DELTAE+ 1.0
C
C  READ INPUT FORCE ITEM CONTRIBUTIONS
    5 IF (ITYP13 .EQ. 0) GO TO 6
      READ (5,107) CAI,CNI,CYI,CLLI,CLMI,CLNI,TYPE
  107 FORMAT (6F10.0,10XI2 )
      IF(TYPE .NE. 13) GO TO 800
      GO TO 60
C  RESET INITIAL ZERO VALUES ON FORCE COEFFICIENTS
    6 CAI = 0.0
      CNI = 0.0
      CYI = 0.0
      CLLI= 0.0
      CLMI= 0.0
      CLNI= 0.0
   60 CONTINUE
C  GO TO CONTROL DEFLECTION SUBROUTINE IF REQUIRED
      IF (IGTYPE .NE. 1 .AND. IGTYPE .NE. 3) GO TO 36
      IF (DELTAE .EQ. DELTAS) GO TO 36
      DELTAS = DELTAE
      CALL CONTRL (DELTAE,ISIZ)
      IF (ERROR .NE. 0) GO TO 806
   36 CONTINUE
C
C  GO TO FORCE CALCULATION ROUTINE
      CALL FORCE (ALPHA,BETA,CPSTAG,SREF,SYMFCT,XCG,YCG,ZCG,MACH,
     1 SPAN,MAC,NABCT,QRP,ALT,PRINT,CAI,CNI,CYI,CLLI,CLMI,CLNI,ETAC,NS,
     2 IMPACT,IPRINT,IFIRST,PSTAG,TSTAG,RENO,ISIZ,ENPM,QQINF,ISHAD,
     3 IORIEN,IMPACI,ISHADI,IDERIV,V,IGTYPE,DELTAE,SWEEP,RETRAN,
     4 HMLT,HMRT,PFS,NW,TWALL)
      IF (ERROR.NE.0) GO TO 806
```

```
DECK AROA

      DELTES(NABCT) = DELTAE
      IMPI(NABCT)= IMPACI
      ISHI(NABCT)= ISHADI
      HML(NABCT) = HMLT
      HMR(NABCT) = HMRT
C
C
      IFLG = IDERIV + 1
      GO TO (70,30,40,50,61,30,40),IFLG
C
C  LONGITUDINAL DERIVATIVES
   30 GO TO (31,32,33),IDSTAT
   31 CASD1= CCA(NABCT)
      CLSD1= CCL(NABCT)
      CNSD1= CCN(NABCT)
      CLMSD1= CCLM(NABCT)
C
      IDSTAT = 2
      ALPHA = ALPHAS
      IF (IDERIV .EQ. 1) GO TO 120
      DELQRP = V*0.5E-4
      QRP = QRP + DELQRP
      GO TO 60
C
   32 CLMSD2 = CCLM(NABCT)
      CASD2  =  CCA(NABCT)
      CNSD2  =  CCN(NABCT)
      QRP    = QRPSS
C
  120 IDSTAT = 3
      GO TO 60
C  CALCULATE LONGITUDINAL DERIVATIVES (PER DEGREE)
   33 DCAA(NABCT)  =  (CASD1 - CCA(NABCT))  /  1.0
      DCLA(NABCT)  =  (CLSD1 - CCL(NABCT))  /  1.0
      DCNA(NABCT)  =  (CNSD1 - CCN(NABCT))  /  1.0
      DCMA(NABCT)  =  (CLMSD1- CCLM(NABCT)) /  1.0
```

```
DECK ARDA
C                                                                    ARDA  2520
      IF (IDERIV .EQ. 1) GO TO 121                                   ARDA  2530
      DCMQ(NABCT)  = ((CLMSD2- CCLM(NABCT))/ DELQRP)*2.0*V/MAC       ARDA  2540
      DCAQ(NABCT)  = ((CASD2 - CCA(NABCT)) / DELQRP)*2.0*V/MAC       ARDA  2550
      DCNQ(NABCT)  = ((CNSD2 - CCN(NABCT)) / DELQRP)*2.0*V/MAC       ARDA  2560
      IDFLGE = 1                                                     ARDA  2570
  121 IDFLGA = 1                                                     ARDA  2580
      GO TO 70                                                       ARDA  2590
C                                                                    ARDA  2600
C  DIRECTIONAL DERIVATIVES                                           ARDA  2610
   40 GO TO (41,42,43),IDSTAT                                        ARDA  2620
   41 CYSD1 = CCY(NABCT)                                             ARDA  2630
      CLNSD1= CCLN(NABCT)                                            ARDA  2640
      CLLSD1= CCLL(NABCT)                                            ARDA  2650
C                                                                    ARDA  2660
      IDSTAT = 2                                                     ARDA  2670
      BETA = BETAS                                                   ARDA  2680
      IF (IDERIV .EQ. 2) GO TO 122                                   ARDA  2690
      DELQRP = V*0.5E-4                                              ARDA  2700
      QRP = QRP + DELQRP                                             ARDA  2710
      GO TO 60                                                       ARDA  2720
C                                                                    ARDA  2730
   42 CYSD2 = CCY(NABCT)                                             ARDA  2740
      CLNSD2= CCLN(NABCT)                                            ARDA  2750
      CLLSD2= CCLL(NABCT)                                            ARDA  2760
C                                                                    ARDA  2770
      QRP    = QRPSS                                                 ARDA  2780
  122 IDSTAT = 3                                                     ARDA  2790
      GO TO 60                                                       ARDA  2800
C  CALCULATE LATERAL-DIRECTIONAL DERIVATIVES (PER DEGREE)            ARDA  2810
   43 DCYB(NABCT)  =  (CYSD1 - CCY(NABCT)) / 1.0                     ARDA  2820
      DCNB(NABCT)  =  (CLNSD1-CCLN(NABCT)) / 1.0                     ARDA  2830
      DCLLB(NABCT) =  (CLLSD1-CCLL(NABCT)) / 1.0                     ARDA  2840
C                                                                    ARDA  2850
      IF (IDERIV .EQ. 2) GO TO 123                                   ARDA  2860
      DCYR(NABCT)  = ((CYSD2 - CCY(NABCT)) / DELQRP)*2.0*V/MAC       ARDA  2870
```

DECK ARDA

```
      DCLNR(NABCT) = ((CLNSD2-CCLN(NABCT)) / DELQRP)*2.0*V/SPAN
      DCLLR(NABCT) = ((CLLSD2-CCLL(NABCT)) / DELQRP)*2.0*V/SPAN
      IDFLGF = 1
  123 IDFLGB = 1
      GO TO 70
C  ROLL DERIVATIVES
   5C WRITE (6,813)
  813 FORMAT (1H ,51H***ROLL DERIVATIVE PART OF PROGRAM IS NOT OPERATIVE
     1 23H AT THE PRESENT TIME*** )
      GO TO 806
C
C  CONTROL SURFACE DERIVATIVES
   61 GO TO (62,63), IDSTAT
   62 CLSD1 = CCL(NABCT)
      CLMSD1= CCLM(NABCT)
      CLLSD1=CCLL(NABCT)
      CYSD1 = CCY(NABCT)
      CLNSD1= CCLN(NABCT)
      CNSD1 = CCN(NABCT)
C
      IDSTAT = 2
      DELTAE = DELTS
      GO TO 60
C  CALCULATE CONTROL SURFACE DERIVATIVES (PER DEGREE)
   63 DCLD(NABCT)  =  (CLSD1  - CCL(NABCT))  / 1.0
      DCMD(NABCT)  =  (CLMSD1 - CCLM(NABCT)) / 1.0
      DCLLD(NABCT) =  (CLLSD1 - CCLL(NABCT)) / 1.0
      DCYD(NABCT)  =  (CYSD1  - CCY(NABCT))  / 1.0
      DCLND(NABCT) =  (CLNSD1 - CCLN(NABCT)) / 1.0
      DCND(NABCT)  =  (CNSD1  - CCN(NABCT))  / 1.0
      IF (ABS(DCLLD(NABCT)).LT.1.0E-10) DCLLD(NABCT) = 0.0
      IF (ABS(DCYD(NABCT)).LT.1.0E-10) DCYD(NABCT) = 0.0
      IF (ABS(DCLND(NABCT)).LT.1.0E-10) DCLND(NABCT) = 0.0
      IDFLGD = 1
C
C
```

```
DECK AROA

C  STEP ALPHA-BETA COUNTER BY ONE                                       AROA  3240
  70  NABCT = NABCT + 1                                                 AROA  3250
C                                                                       AROA  3260
C  CHECK ON ALPHA-BETA COUNTER FOR END OF ALPHA-BETA SET                AROA  3270
      IF (NABCT.LE.NAB) GO TO 2                                         AROA  3280
C                                                                       AROA  3290
C   CALCULATE PLUNGE DERIVATIVES IF REQUIRED                            AROA  3300
  21  IF (IABDOT .EQ. 0) GO TO 125                                      AROA  3310
      DO 124 J=1,NAB                                                    AROA  3320
      IF (IDERIV.NE.5 .AND. IDERIV.NE.6) GO TO 124                      AROA  3330
      CALL PLUNGE (IDERS(NAB),DCMA(NAB),DCYB(NAB),DCMADT(NAB),          AROA  3340
     1  DCYBDT(NAB))                                                    AROA  3350
      IF (ERROR .NE. 0) GO TO 800                                       AROA  3360
 124  CONTINUE                                                          AROA  3370
C                                                                       AROA  3380
C   CALCULATE EFFECT OF INPUT VECTORS IF REQUIRED                       AROA  3390
 125  IF (IVECT .EQ. 0) GO TO 127                                       AROA  3400
      DO 126 J=1,NAB                                                    AROA  3410
      CALL VECTOR (MACH,PFS,SREF,XCG,YCG,ZCG,SPAN,MAC,                  AROA  3420
     1  ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCN(J),BET(J),               AROA  3430
     2  CLOD(J),CCLM(J),CCLL(J),CCLN(J))                                AROA  3440
      IF (ERROR .NE. 0) GO TO 800                                       AROA  3450
 126  CONTINUE                                                          AROA  3460
C  WRITE OUT SUMMARY OF FORCE DATA                                      AROA  3470
 127  DO 14 J=1,NAB                                                     AROA  3480
      IF (NPRT.GE.28) GO TO 3                                           AROA  3490
      NPRT = NPRT + 2                                                   AROA  3500
       GO TO 4                                                          AROA  3510
   3  NPRT = 0                                                          AROA  3520
      CALL HEADER                                                       AROA  3530
      WRITE (6,105) MACH,V,RENO                                         AROA  3540
 105  FORMAT (1H0,7H  MACH=F8.3,6H  VEL=F9.1,16H FT/SEC  RE/FT =E13.5 ) AROA  3550
      IF (PSTAG .LT. 0.00001) WRITE (6,111) ALT                         AROA  3560
 111  FORMAT (1H ,7H  ALT =F8.0 )                                       AROA  3570
      IF (PSTAG .GT. 0.00001) WRITE (6,112) PSTAG,TSTAG                 AROA  3580
 112  FORMAT (1H ,16X7HP STAG=F7.1,16H ATMOS   T STAG=F7.1,6H DEG F )   AROA  3590
```


AERO

DECK AROA

```
      WRITE (6,113) SREF,SPAN,MAC,XCG,YCG,ZCG
  113 FORMAT (1H0,9H  S REF =F9.2,8H  SPAN =F8.2,8H  MAC  =F8.2,/1H ,
     1  9H  X CG  =F9.2,8H  Y CG =F8.2,8H  Z CG =F8.2 )
      WRITE (6,114)
  114 FORMAT (1H0,10HFORCE DATA,71X12HCONTROL DATA,/1H ,
     1 7H  ALPHA,4X3HC D,7X3HC L,7X3HC A,7X3HC Y,7X3HC N,6X3H  K,
     2 20X29HIMPACT  ETAC  IMPACI  DELTA E,/1H ,
     3 6H  BETA,5X3HL/D,7X3HC M,7X4HC LL,6X4HC LN,6X3HC F,6X7HQ/Q INF,
     4 16X20HISHAD   ENPM  ISHADI  )
C
      IF (IDFLGA .EQ. 1) WRITE (6,807)
  8C7 FORMAT (1H ,11X4HCA A,6X4HCL A,6X4HCN A,6X4HCM A)
      IF (IDFLGB .EQ. 1) WRITE (6,808)
  8C8 FORMAT (1H ,11X4HCY B,6X5HCLN B,5X5HCLL B)
      IF (IDFLGE .EQ. 1) WRITE (6,816)
  816 FORMAT (1H ,11X4HCM Q,6X4HCA Q,6X4HCN Q,6X5HCMA D,39X1HQ )
      IF (IDFLGF .EQ. 1) WRITE (6,817)
  817 FORMAT (1H ,11X4HCY R,6X5HCLN R,5X5HCLL R,5X5HCYB D,39X1HR )
      IF (IDFLGD .EQ. 1) WRITE (6,814)
  814 FORMAT (1H ,11X4HCM D,6X5HCLL D,5X4HCY D,6X5HCLN D,5X4HCN D,29X,
     1  4HHM L,4X4HHM R)
C
    4 WRITE (6,106)ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCN(J),CPS(J),
     1 IMP(J),ETACS(J),IMPI(J),DELTES(J),
     2 BET(J),CLOD(J),CCLM(J),CCLL(J),CCLN(J),CF(J),QQINFS(J),
     3 ISH(J),ENPMS(J),ISHI(J)
  1C6 FORMAT (1H0,F7.2,6F10.5,16XI3,F10.4,I5,F10.2,/1H ,F7.2,6F10.5,
     1 16XI3,F10.4,I5 )
      IF (IDFLGA .EQ. 1) WRITE (6,809) DCAA(J),DCLA(J),DCNA(J),DCMA(J)
  809 FORMAT (1H ,7X,4F10.5)
      IF (IDFLGB .EQ. 1) WRITE (6,812) DCYB(J),DCNB(J),DCLLB(J)
  812 FORMAT (1H ,7X,3F10.5)
      IF (IDFLGE .EQ. 1) WRITE (6,818) DCMQ(J),DCAQ(J),DCNQ(J),
     1                         DCMADT(J),QRPS(J)
  818 FORMAT (1H ,7X,4F10.5,34X,1PE10.3)
      IF (IDFLGF .EQ. 1) WRITE (6,819) DCYR(J),DCLNR(J),DCLLR(J),
```

DECK AROA

```
      1                        DCYBDT(J),QRPS(J)                            AROA
  819 FORMAT (1H ,7X,4F10.5,34X,1PE10.3)                                    AROA
      IF (IDFLGD .EQ. 1) WRITE (6,815) DCMD(J),DCLLD(J),DCYD(J),            AROA
     1  DCLND(J),DCND(J),HML(J),HMR(J)                                      AROA
  815 FORMAT (1H ,8X,1PE10.2,4E10.2,22X,2E10.2)                             AROA
      IF (IDFLGA.EQ.1) NPRT = NPRT + 1                                      AROA
      IF (IDFLGB.EQ.1) NPRT = NPRT + 1                                      AROA
      IF (IDFLGD.EQ.1) NPRT = NPRT + 1                                      AROA
      IF (IDFLGE.EQ.1) NPRT = NPRT + 1                                      AROA
      IF (IDFLGF.EQ.1) NPRT = NPRT + 1                                      AROA
C                                                                           AROA
      IF (IPS .EQ. 0) GO TO 20                                              AROA
      WRITE (9) ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCN(J),BET(J),           AROA
     1          CLOD(J),CCLM(J),CCLL(J),CCLN(J),CF(J),IPL9                  AROA
C  STORE DATA FOR FUTURE SUMMATION ON TAPE 1 IF REQUIRED                    AROA
   20 IF (ISUM .NE. 1) GO TO 14                                             AROA
      WRITE (1) ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCN(J),CPS(J),           AROA
     1 BET(J),CLOD(J),CCLM(J),CCLL(J),CCLN(J),CF(J),QQINFS(J),              AROA
     2 DCAA(J),DCLA(J),DCNA(J),DCMA(J),DCMQ(J),DCAQ(J),DCNQ(J),             AROA
     3 DCYB(J),DCNB(J),DCLLB(J),DCYR(J),DCLNR(J),DCLLR(J),DELTES(J),        AROA
     4 DCLD(J),DCMD(J),DCLLD(J),DCYD(J),DCLND(J),DCND(J),HML(J),HMR(J),     AROA
     5 DCMADT(J),DCYBDT(J)                                                  AROA
      NSAVE = NSAVE + 1                                                     AROA
   14 CONTINUE                                                              AROA
      IF (ISUM .EQ. 1) WRITE (6,128)                                        AROA
  128 FORMAT (1H0,10X,40HTHESE DATA HAVE BEEN SAVED FOR SUMMATION)          AROA
      IF (ISUM .GE. 2) WRITE (6,130) NSAVE2                                 AROA
  130 FORMAT (1H0,10X,31HTHESE DATA ARE THE SUMMATION OF,I4,                AROA
     1 28H COMPONENTS PREVIOUSLY SAVED )                                    AROA
C                                                                           AROA
      IF (IPS .EQ. 0) GO TO 16                                              AROA
      WRITE (10) NAB,NAB,NAB,NAB,IPL10                                      AROA
      WRITE (6,129)                                                         AROA
  129 FORMAT (1H0,10X,39HTHESE DATA HAVE BEEN SAVED FOR PLOTTING )          AROA
   16 CONTINUE                                                              AROA
C                                                                           AROA
```

```
DECK AROA

C  CHECK IF LAST MACH-ALT CONDITION HAS BEEN CALCULATED
  10  IF (LAST) 11,1,11
C
C  ADD UP PREVIOUSLY SAVED DATA AND PRINT (IF ISUM = 2)
  15  REWIND 1
      NSAVE2 = NSAVE / NAB
      DO 17 I=1,NSAVE2
       DO 18 J=1,NAB
        READ (1) ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCN(J),CPS(J),
     1    BET(J),CLOD(J),CCLM(J),CCLL(J),CCLN(J),CF(J),QQINFS(J),
     2    DCAA(J),DCLA(J),DCNA(J),DCMA(J),DCMQ(J),DCAQ(J),DCNQ(J),
     3    DCYB(J),DCNB(J),DCLLB(J),DCYR(J),DCLNR(J),DCLLR(J),DELTES(J),
     4    DCLD(J),DCMD(J),DCLLD(J),DCYD(J),DCLND(J),DCND(J),HML(J),
     5    HMR(J),DCMADT(J),DCYBDT(J)
          CCDS(J)= CCDS(J) + CCD(J)
          CCLS(J)= CCLS(J) + CCL(J)
          CCAS(J)= CCAS(J) + CCA(J)
          CCYS(J)= CCYS(J) + CCY(J)
          CCNS(J)= CCNS(J) + CCN(J)
          CCLMS(J) = CCLMS(J) + CCLM(J)
          CCLLS(J) = CCLLS(J) + CCLL(J)
          CCLNS(J) = CCLNS(J) + CCLN(J)
          CFS(J) = CFS(J) + CF(J)
          DCAAS(J) =  DCAAS(J) + DCAA(J)
          DCLAS(J) =  DCLAS(J) + DCLA(J)
          DCNAS(J) =  DCNAS(J) + DCNA(J)
          DCMAS(J) =  DCMAS(J) + DCMA(J)
          DCMQS(J) =  DCMQS(J) + DCMQ(J)
          DCAQS(J) =  DCAQS(J) + DCAQ(J)
          DCNQS(J) =  DCNQS(J) + DCNQ(J)
          DCYBS(J) =  DCYBS(J) + DCYB(J)
          DCNBS(J) =  DCNBS(J) + DCNB(J)
          DCLLBS(J)=  DCLLBS(J)+ DCLLB(J)
          DCYRS(J) =  DCYRS(J) + DCYR(J)
          DCLNRS(J)=  DCLNRS(J)+ DCLNR(J)
          DCLLRS(J)=  DCLLRS(J)+ DCLLR(J)
```

DECK AROA

```
            DCLDS(J) =  DCLDS(J) + DCLD(J)
            DCMDS(J) =  DCMDS(J) + DCMD(J)
            DCLLDS(J)=  DCLLDS(J)+ DCLLD(J)
            DCYDS(J) =  DCYDS(J) + DCYD(J)
            DCLNDS(J)=  DCLNDS(J)+ DCLND(J)
            DCNDS(J) =  DCNDS(J) + DCND(J)
            HMLS(J)  =  HMLS(J)  + HML(J)
            HMRS(J)  =  HMRS(J)  + HMR(J)
            DCMADS(J)=  DCMADS(J)+ DCMADT(J)
            DCYBDS(J)=  DCYBDS(J)+ DCYBDT(J)
   18   CONTINUE
   17  CONTINUE
C
       DO 19 J=1,NAB
        CCD(J) = CCDS(J)
        CCL(J) = CCLS(J)
        CCA(J) = CCAS(J)
        CCY(J) = CCYS(J)
        CCN(J) = CCNS(J)
        CCLM(J) = CCLMS(J)
        CCLL(J) = CCLLS(J)
        CCLN(J) = CCLNS(J)
        CF(J) = CFS(J)
        DCAA(J) = DCAAS(J)
        DCLA(J) = DCLAS(J)
        DCNA(J) = DCNAS(J)
        DCMA(J) = DCMAS(J)
        DCMQ(J) = DCMQS(J)
        DCAQ(J) = DCAQS(J)
        DCNQ(J) = DCNQS(J)
        DCYB(J) = DCYBS(J)
        DCNB(J) = DCNBS(J)
        DCLLB(J)= DCLLBS(J)
        DCYR(J) = DCYRS(J)
        DCLNR(J)= DCLNRS(J)
        DCLLR(J)= DCLLRS(J)
```

DECK AROA

```
      DCLD(J)  = DCLDS(J)                                        AROA  5040
      DCMD(J)  = DCMDS(J)                                        AROA  5050
      DCLLD(J)= DCLLDS(J)                                        AROA  5060
      DCYD(J)  = DCYDS(J)                                        AROA  5070
      DCLND(J)= DCLNDS(J)                                        AROA  5080
      DCND(J)  = DCNDS(J)                                        AROA  5090
      HML(J)   = HMLS(J)                                         AROA  5100
      HMR(J)   = HMRS(J)                                         AROA  5110
      DCMADT(J)= DCMADS(J)                                       AROA  5120
      DCYBDT(J)= DCYBDS(J)                                       AROA  5130
       CCDS(J)  = 0.0                                            AROA  5140
       CCLS(J)  = 0.0                                            AROA  5150
       CCAS(J)  = 0.0                                            AROA  5160
       CCYS(J)  = 0.0                                            AROA  5170
       CCNS(J)  = 0.0                                            AROA  5180
       CCLMS(J)= C.0                                             AROA  5190
       CCLLS(J)= 0.0                                             AROA  5200
       CCLNS(J)= 0.0                                             AROA  5210
       CFS(J)   = 0.0                                            AROA  5220
       DCAAS(J)  = 0.0                                           AROA  5230
       DCLAS(J)  = 0.0                                           AROA  5240
       DCNAS(J)  = 0.0                                           AROA  5250
       DCMAS(J)  = 0.0                                           AROA  5260
       DCMQS(J)  = 0.0                                           AROA  5270
       DCAQS(J)  = 0.0                                           AROA  5280
       DCNQS(J)  = 0.0                                           AROA  5290
       DCYBS(J)  = 0.0                                           AROA  5300
       DCNBS(J)  = 0.0                                           AROA  5310
       DCLLBS(J)= 0.0                                            AROA  5320
       DCYRS(J)  = 0.0                                           AROA  5330
       DCLNRS(J)= 0.0                                            AROA  5340
       DCLLRS(J)= 0.C                                            AROA  5350
       DCLDS(J)  = 0.0                                           AROA  5360
       DCMDS(J)  = 0.0                                           AROA  5370
       DCLLDS(J)= 0.0                                            AROA  5380
       DCYDS(J)  = 0.0                                           AROA  5390
```

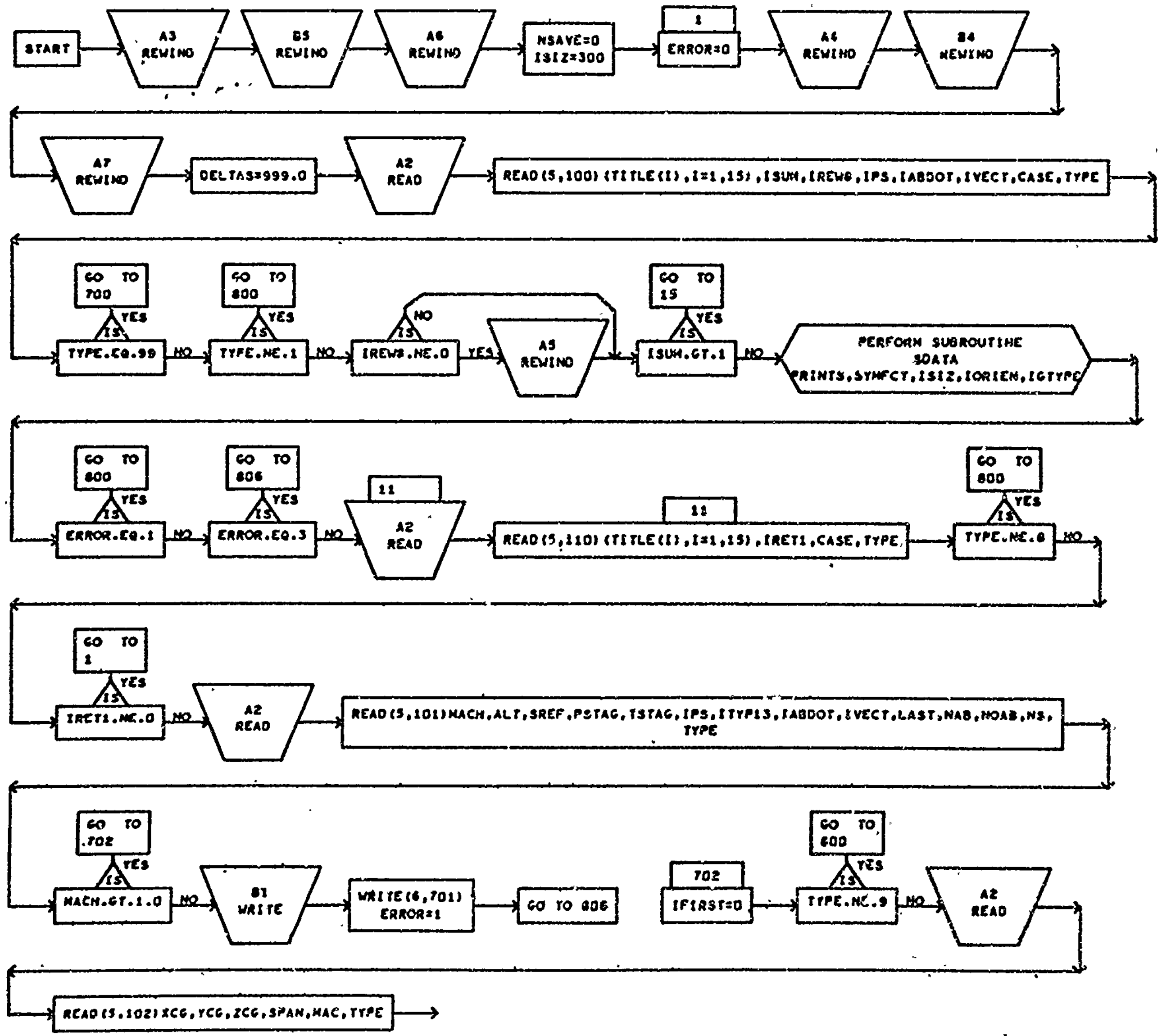
DECK AROA

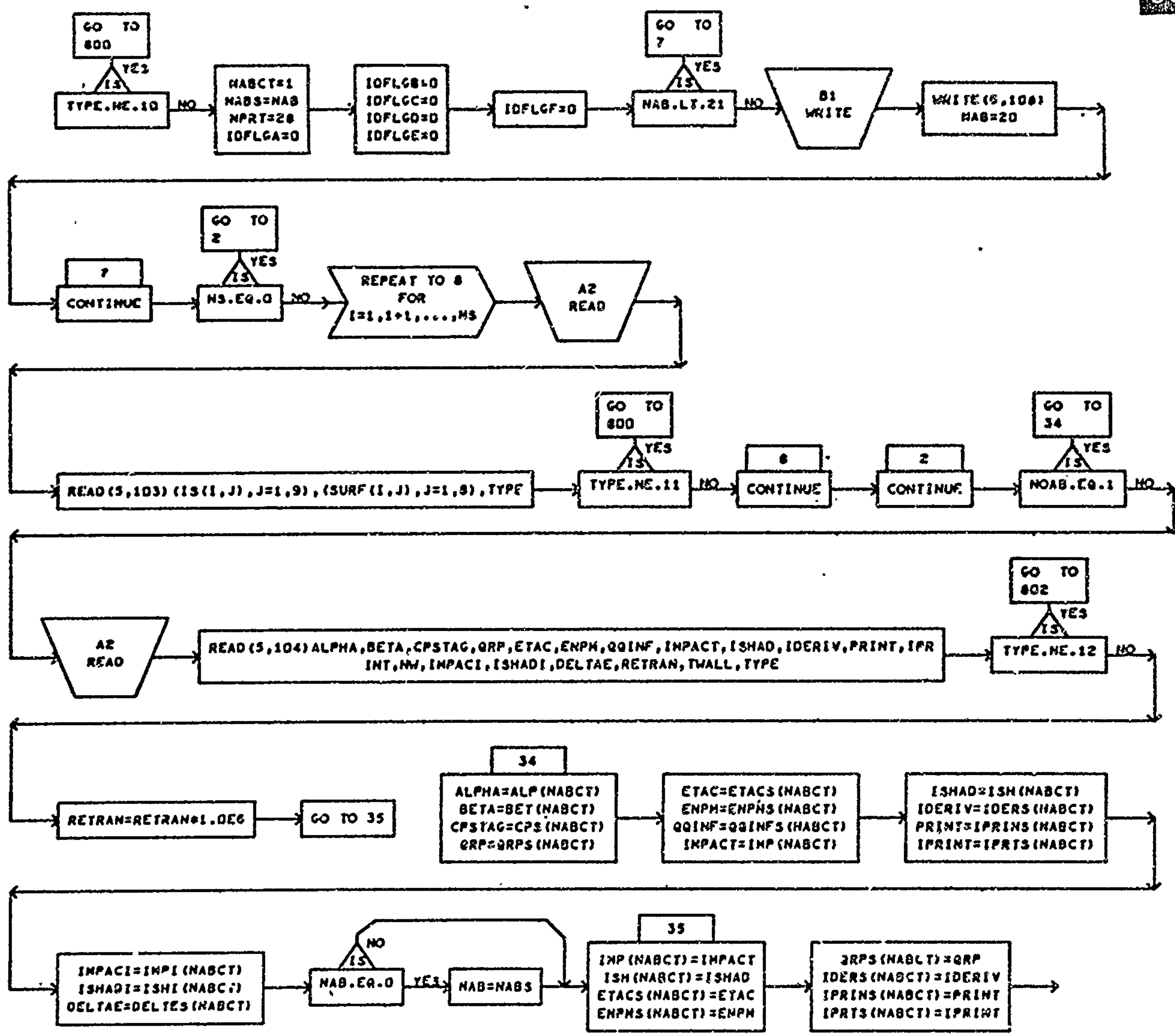
```
        DCLNDS(J)= 0.0                                                   AROA  5400
        DCNDS(J) = 0.0                                                   AROA  5410
       HMLS(J)   = 0.0                                                   AROA  5420
       DCMADS(J) = 0.0                                                   AROA  5430
       DCYBDS(J) = 0.0                                                   AROA  5440
       HMRS(J)   = 0.0                                                   AROA  5450
   19  CLOD(J) = CCL(J) / CCD(J)                                         AROA  5460
C                                                                        AROA  5470
      LAST = C                                                           AROA  5480
      NPRT = 28                                                          AROA  5490
C  CHECK ISUM FLAG AND SET TAPE 1 TO PROPER POSITION FOR NEXT RUN        AROA  5500
C     IF ISUM = 2 LEAVE TAPE 1 IN ITS PRESENT POSITION FOR NEW DATA      AROA  5510
C     IF ISUM = 3 REWIND TAPE 1 FOR A NEW SET OF SAVED DATA              AROA  5520
C     IF ISUM = 4 BACKSPACE TAPE 1 ONE SET OF SUMMATION DATA             AROA  5530
C     IF ISUM = 5 AFTER PRINTING OF SUMMATION DATA REWINC SUMMATION TAPE AROA  5540
C               AND WRITE THE SUMMATICN DATA ONLY ON THE TAPE            AROA  5550
      GO TO (21,21,22,23,25),ISUM                                        AROA  5560
   22 REWIND 1                                                           AROA  5570
      NSAVE = 0                                                          AROA  5580
      GO TO 21                                                           AROA  5590
   23 DO 24 J=1,NAB                                                      AROA  5600
   24 BACKSPACE 1                                                        AROA  5610
      NSAVE = NSAVE - NAB                                                AROA  5620
      GO TO 21                                                           AROA  5630
   25 REWIND 1                                                           AROA  5640
      NSAVE = 0                                                          AROA  5650
      ISUM  = 1                                                          AROA  5660
      GO TO 21                                                           AROA  5670
C                                                                        AROA  5680
C  ERROR CHECK ON READING CARDS                                          AROA  5690
  802 WRITE (6,804)                                                      AROA  5700
  8C4 FORMAT (1HO,48H***** PRCGRAM HAS ATTEMPTED TO READ A ALPHA-BETA    AROA  5710
     147H COMBINATION CARD WITH THE WRONG TYPE CODE***** )               AROA  5720
C                                                                        AROA  5730
C  WRITE OUT CARD TYPE ERROR STATEMENT AND TERMINATE JOB                 AROA  5740
  800 WRITE (6,801)                                                      AROA  5750
```

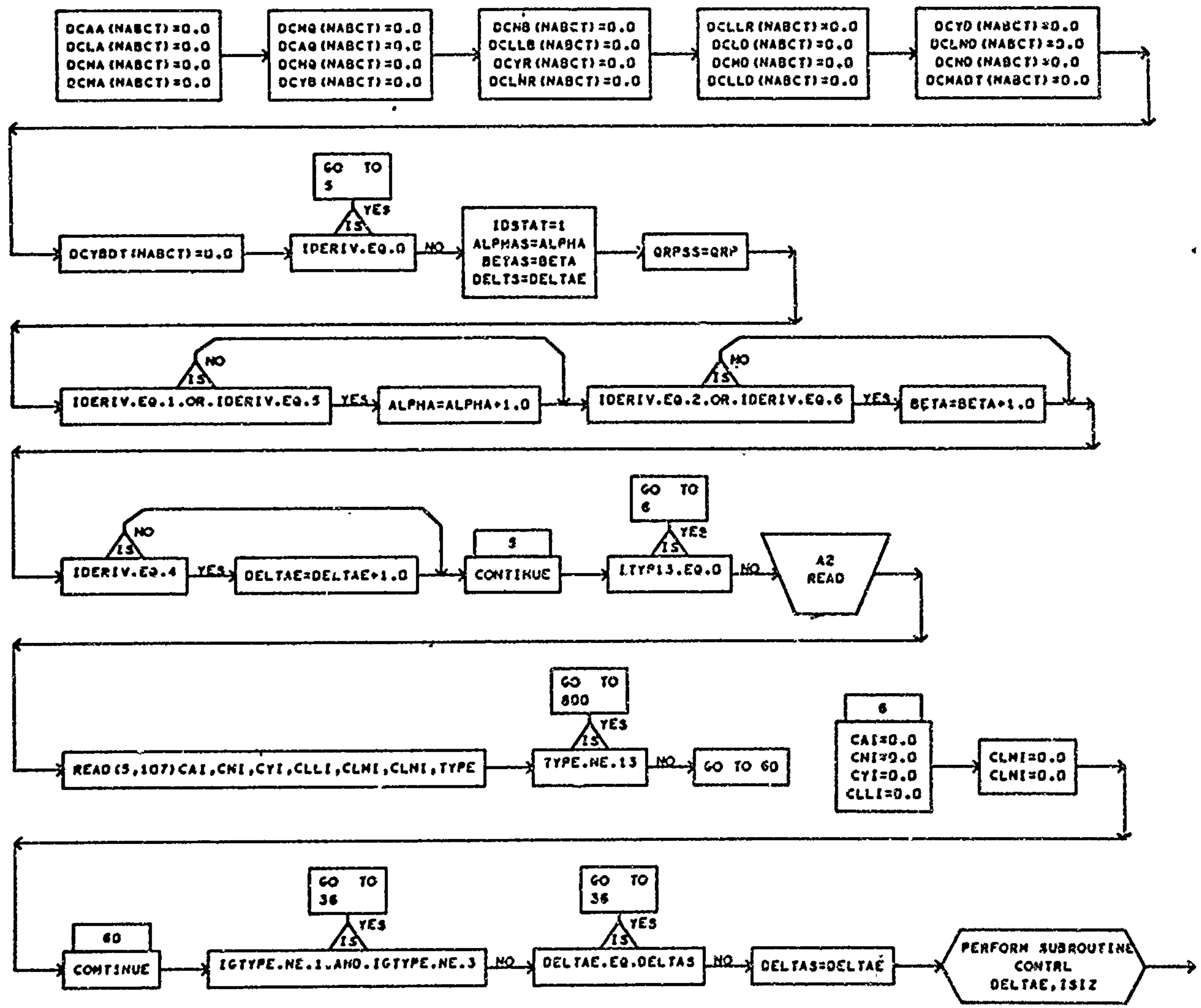

AERO

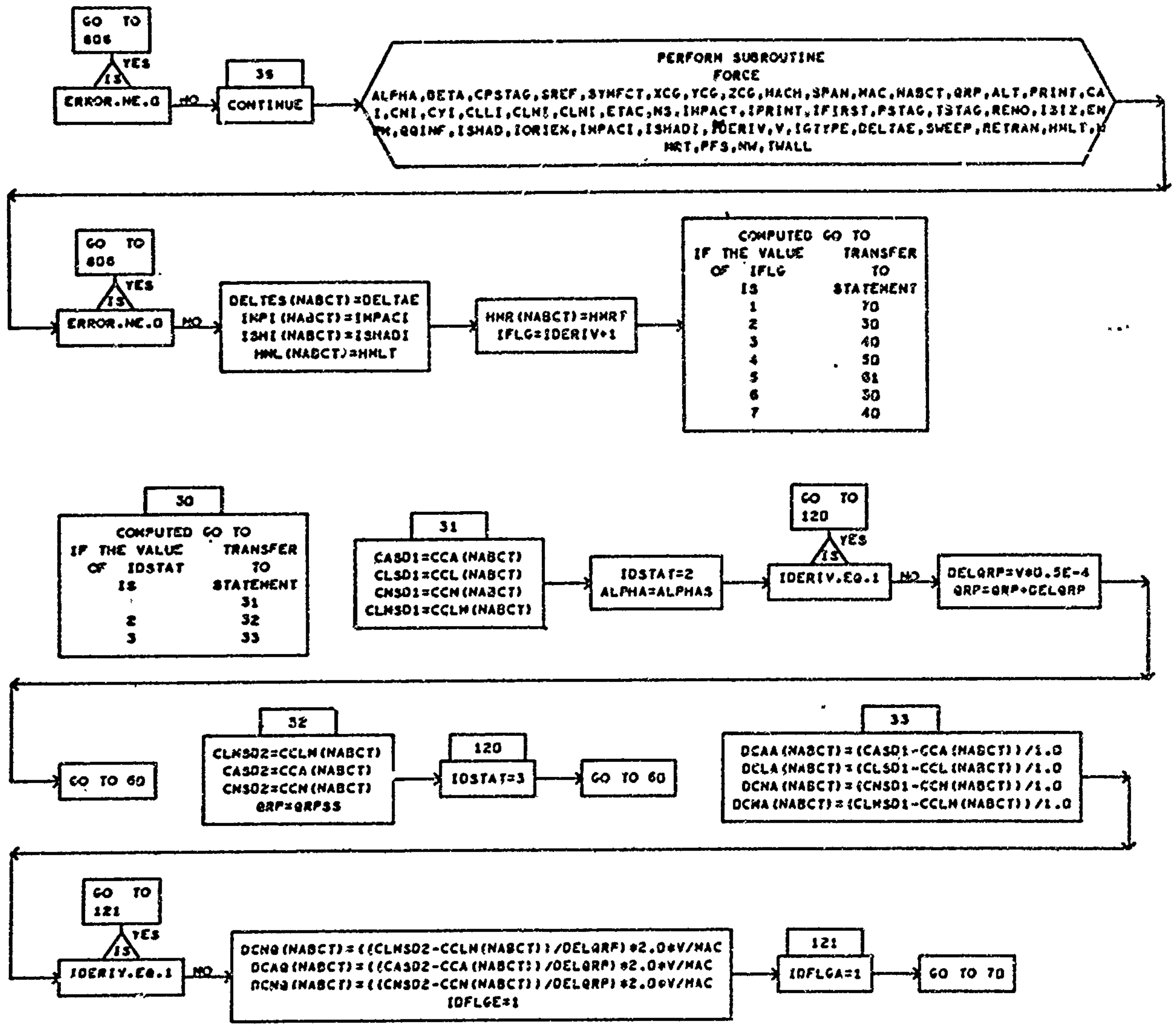
```
CECK AROA

 801  FORMAT (1HO,51H **************************************************   AROA  5760
     163H**************************************************************   AROA  5770
     2 /1H ,   54H*** CONGRATULATIONS - YOU HAVE HIT THE JACKPOT WITH AN   AROA  5780
     3 61H ERROR INVOLVING EITHER CARD ORDER OR CARD TYPE INDICATION***)   AROA  5790
      READ (5,810) (CARD(I),I=1,20)                                       AROA  5800
 810  FORMAT (20A4)                                                       AROA  5810
      WRITE (6,805) (CARD(I),I=1,20)                                      AROA  5820
 805  FORMAT (1HO,45H THE CARD LOCATED JUST BEFORE THE CARD LISTED        AROA  5830
     1 18H BELOW IS IN ERROR,/1H 20A4)                                    AROA  5840
      ERROR = 1                                                           AROA  5850
 806  RETURN                                                              AROA  5860
 700  ERROR = 0                                                           AROA  5870
      RETURN                                                              AROA  5880
C                                                                         AROA  5890
      END                                                                 AROA  5900
```

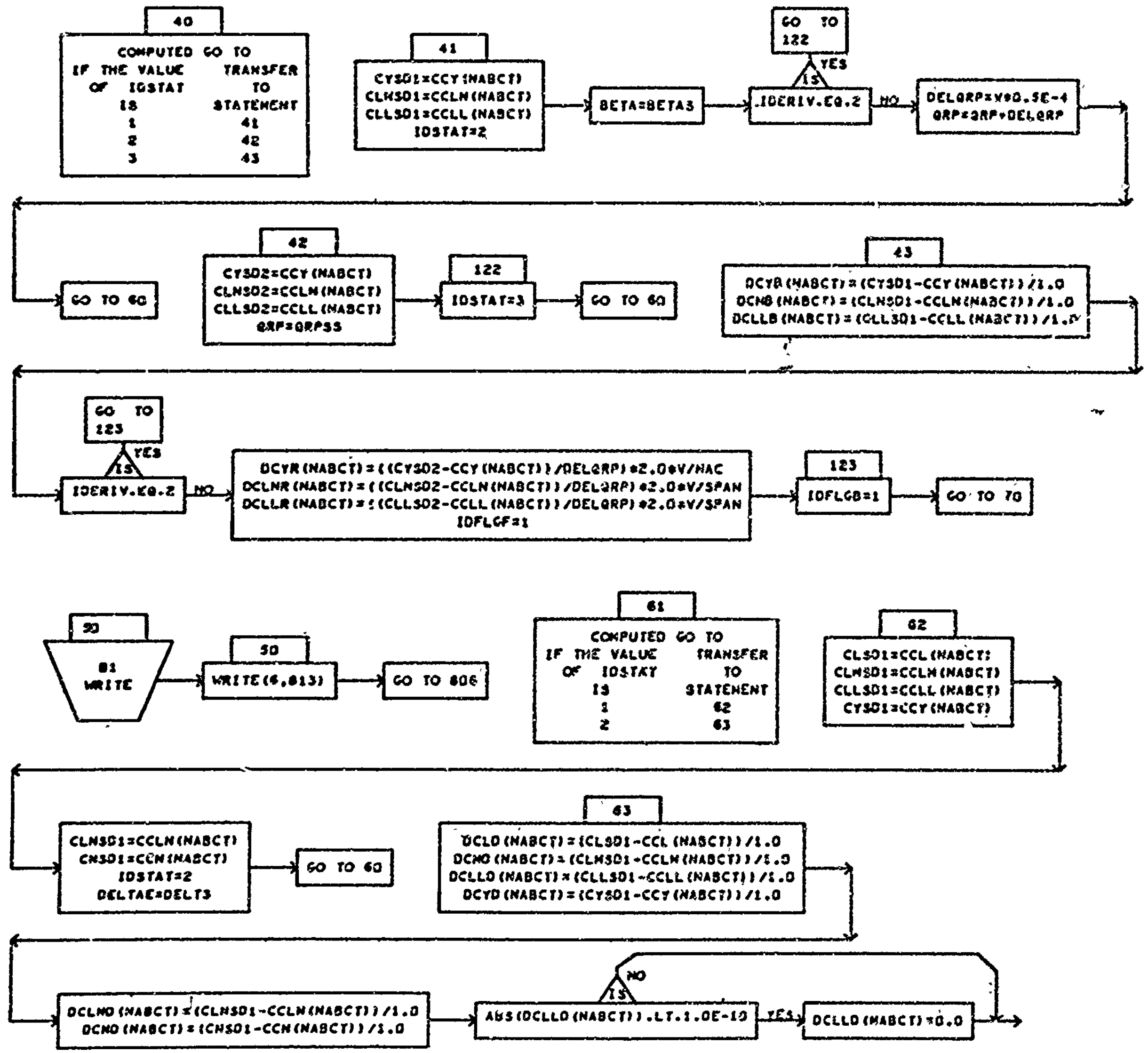
SUBROUTINE AERO



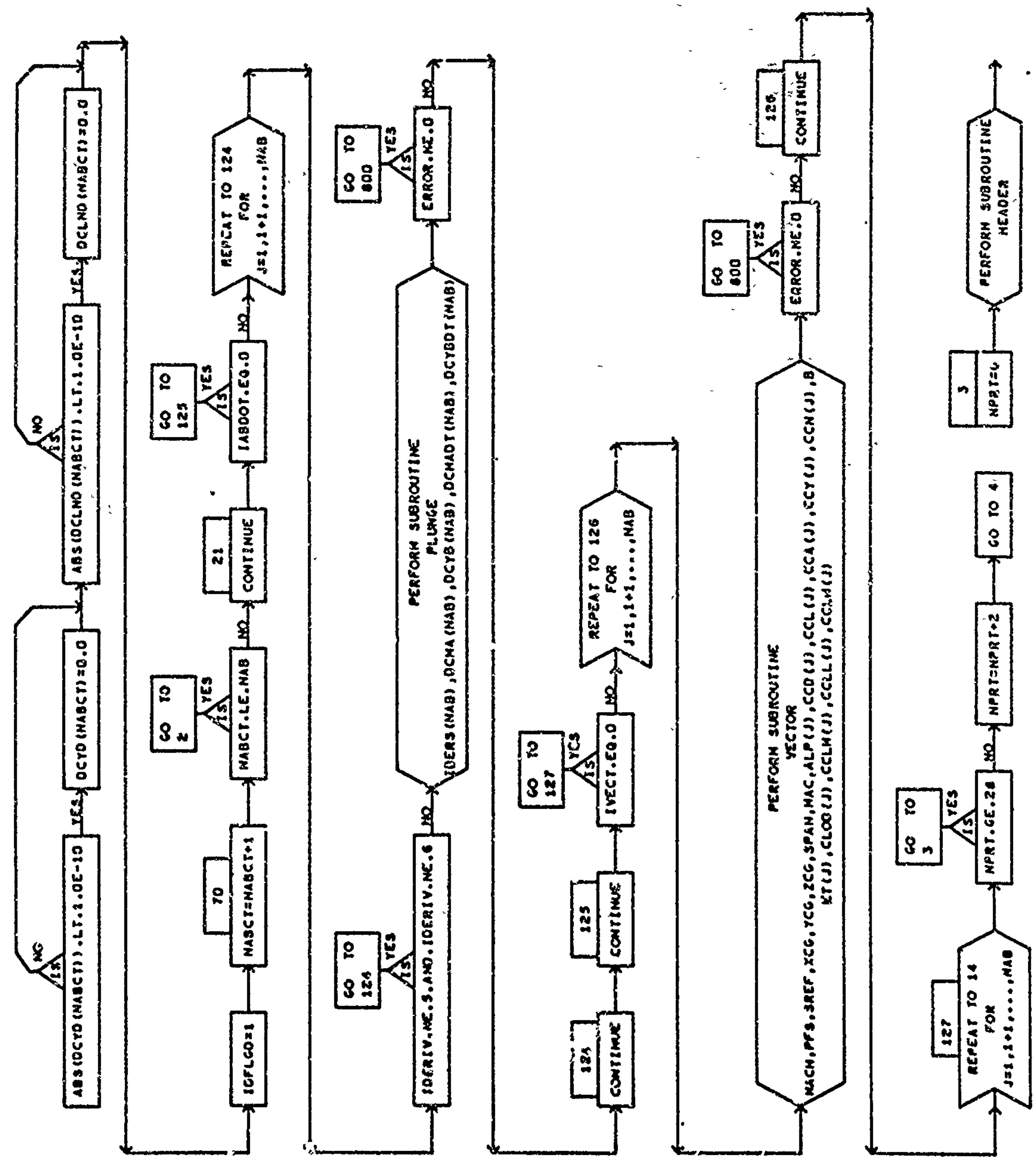

GO TO 800
YES
IS
TYPE.NE.10
NO
NABCT=1
NABS=NAB
NPRT=28
IDFLGA=0
IDFLGB=0
IDFLGC=0
IDFLGD=0
IDFLGE=0
IDFLGF=0
GO TO 7
YES
IS
NAB.LT.21
NO
B1
WRITE
WRITE(6,108)
NAB=20
7
CONTINUE
GO TO 2
YES
IS
NS.EQ.0
NO
REPEAT TO 8
FOR
I=1,1+1,....,NS
A2
READ
READ(5,103) (IS(I,J),J=1,9),(SURF(I,J),J=1,8),TYPE
GO TO 800
YES
IS
TYPE.NE.11
NO
8
CONTINUE
2
CONTINUE
GO TO 34
YES
IS
NOAB.EQ.1
NO
A2
READ
READ(5,104) ALPHA,BETA,CPSTAG,QRP,ETAC,ENPH,QQINF,IMPACT,ISHAD,IDERIV,PRINT,IPR
INT,NW,IMPACI,ISHADI,DELTAE,RETRAN,TWALL,TYPE
GO TO 802
YES
IS
TYPE.NE.12
NO
RETRAN=RETRAN*1.0E6
GO TO 35
34
ALPHA=ALP(NABCT)
BETA=BET(NABCT)
CPSTAG=CPS(NABCT)
QRP=QRPS(NABCT)
ETAC=ETACS(NABCT)
ENPH=ENPHS(NABCT)
QQINF=QQINFS(NABCT)
IMPACT=IMP(NABCT)
ISHAD=ISH(NABCT)
IDERIV=IDERS(NABCT)
PRINT=IPRINS(NABCT)
IPRINT=IPRTS(NABCT)
IMPACI=IMPI(NABCT)
ISHADI=ISHI(NABCT)
DELTAE=DELTES(NABCT)
NO
IS
NAB.EQ.0
YES
NAB=NABS
35
IMP(NABCT)=IMPACT
ISH(NABCT)=ISHAD
ETACS(NABCT)=ETAC
ENPHS(NABCT)=ENPH
QRPS(NABCT)=QRP
IDERS(NABCT)=IDERIV
IPRINS(NABCT)=PRINT
IPRTS(NABCT)=IPRINT


DCAA(NABCT)=0.0
DCLA(NABCT)=0.0
DCNA(NABCT)=0.0
DCMA(NABCT)=0.0
DCMQ(NABCT)=0.0
DCAQ(NABCT)=0.0
DCNQ(NABCT)=0.0
DCYB(NABCT)=0.0
DCNB(NABCT)=0.0
DCLLB(NABCT)=0.0
DCYR(NABCT)=0.0
DCLNR(NABCT)=0.0
DCLLR(NABCT)=0.0
DCLO(NABCT)=0.0
DCMO(NABCT)=0.0
DCLLD(NABCT)=0.0
DCYD(NABCT)=0.0
DCLND(NABCT)=0.0
DCND(NABCT)=0.0
DCMADT(NABCT)=0.0
DCYBDT(NABCT)=0.0
IS IDERIV.EQ.0
YES
GO TO 5
NO
IDSTAT=1
ALPHAS=ALPHA
BETAS=BETA
DELTS=DELTAE
QRPSS=QRP
IS IDERIV.EQ.1.OR.IDERIV.EQ.5
NO
YES
ALPHA=ALPHA+1.0
IS IDERIV.EQ.2.OR.IDERIV.EQ.6
NO
YES
BETA=BETA+1.0
IS IDERIV.EQ.4
NO
YES
DELTAE=DELTAE+1.0
5
CONTINUE
IS LTYP13.EQ.0
YES
GO TO 6
NO
A2
READ
READ(5,107)CAI,CNI,CYI,CLLI,CLNI,CLNI,TYPE
IS TYPE.NE.13
YES
GO TO 800
NO
GO TO 60
6
CAI=0.0
CNI=0.0
CYI=0.0
CLLI=0.0
CLNI=0.0
CLNI=0.0
60
CONTINUE
IS IGTYPE.NE.1.AND.IGTYPE.NE.3
YES
GO TO 36
NO
IS DELTAE.EQ.DELTAS
YES
GO TO 36
NO
DELTAS=DELTAE
PERFORM SUBROUTINE
CONTRL
DELTAE,ISIZ

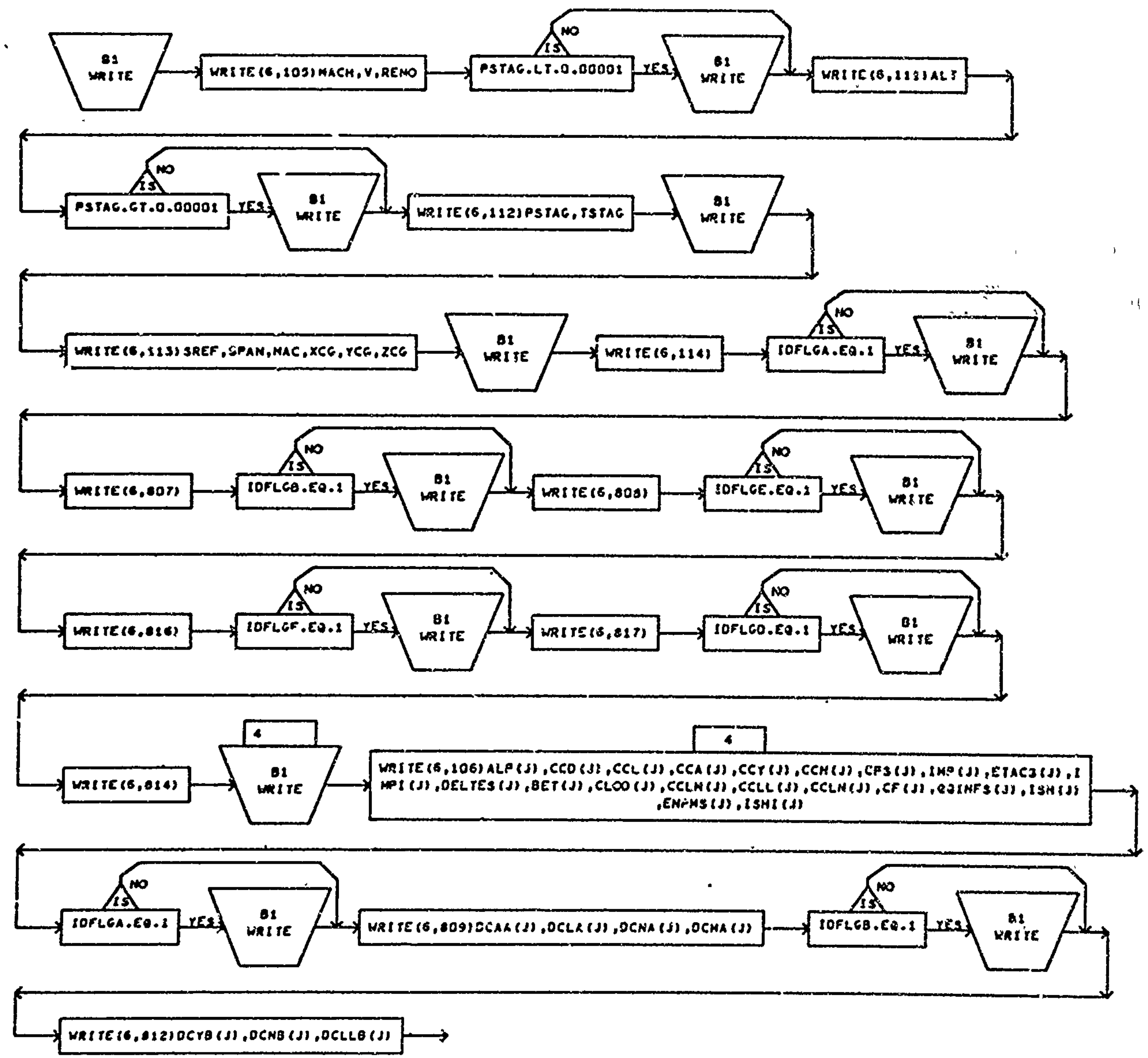

GO TO 806
YES
IS
ERROR.NE.0
NO
35
CONTINUE
PERFORM SUBROUTINE
FORCE
ALPHA,BETA,CPSTAG,SREF,SYMFCT,XCG,YCG,ZCG,MACH,SPAN,MAC,NABCT,QRP,ALT,PRINT,CA
I,CNI,CYI,CLLI,CLMI,CLNI,ETAC,NS,IMPACT,IPRINT,IFIRST,PSTAG,TSTAG,RENO,ISIZ,EM
M,QQINF,ISHAD,IORIEN,IMPACI,ISHADI,IDERIV,V,IGTYPE,DELTAE,SWEEP,RETRAN,HHLT,H
HRT,PFS,NW,TWALL
GO TO 806
YES
IS
ERROR.NE.0
NO
DELTES(NABCT)=DELTAE
IMPI(NABCT)=IMPACI
ISHI(NABCT)=ISHADI
HHL(NABCT)=HHLT
HHR(NABCT)=HHRT
IFLG=IDERIV+1
COMPUTED GO TO
IF THE VALUE TRANSFER
OF IFLG TO
IS STATEMENT
1 70
2 30
3 40
4 50
5 61
6 30
7 40
30
COMPUTED GO TO
IF THE VALUE TRANSFER
OF IDSTAT TO
IS STATEMENT
1 31
2 32
3 33
31
CASD1=CCA(NABCT)
CLSD1=CCL(NABCT)
CNSD1=CCN(NABCT)
CLMSD1=CCLM(NABCT)
IDSTAT=2
ALPHA=ALPHAS
GO TO 120
YES
IS
IDERIV.EQ.1
NO
DELQRP=V*0.5E-4
QRP=QRP+DELQRP
GO TO 60
32
CLMSD2=CCLM(NABCT)
CASD2=CCA(NABCT)
CNSD2=CCN(NABCT)
QRP=QRPSS
120
IDSTAT=3
GO TO 60
33
DCAA(NABCT)=(CASD1-CCA(NABCT))/1.0
DCLA(NABCT)=(CLSD1-CCL(NABCT))/1.0
DCNA(NABCT)=(CNSD1-CCN(NABCT))/1.0
DCMA(NABCT)=(CLMSD1-CCLM(NABCT))/1.0
GO TO 121
YES
IS
IDERIV.EQ.1
NO
DCMQ(NABCT)=((CLMSD2-CCLM(NABCT))/DELQRF)*2.0*V/MAC
DCAQ(NABCT)=((CASD2-CCA(NABCT))/DELQRP)*2.0*V/MAC
DCNQ(NABCT)=((CNSD2-CCN(NABCT))/DELQRP)*2.0*V/MAC
IDFLGE=1
121
IDFLGA=1
GO TO 70


40
COMPUTED GO TO
IF THE VALUE OF IDSTAT IS 1 2 3
TRANSFER TO STATEMENT 41 42 43
41
CYSD1=CCY(NABCT)
CLNSD1=CCLN(NABCT)
CLLSD1=CCLL(NABCT)
IDSTAT=2
BETA=BETAS
IS
IDERIV.EQ.2
YES
GO TO 122
NO
DELQRP=V*0.5E-4
QRP=QRP+DELQRP
GO TO 60
42
CYSD2=CCY(NABCT)
CLNSD2=CCLN(NABCT)
CLLSD2=CCLL(NABCT)
QRP=QRPSS
122
IDSTAT=3
GO TO 60
43
DCYB(NABCT)=(CYSD1-CCY(NABCT))/1.0
DCNB(NABCT)=(CLNSD1-CCLN(NABCT))/1.0
DCLLB(NABCT)=(CLLSD1-CCLL(NABCT))/1.0
IS
IDERIV.EQ.2
YES
GO TO 123
NO
DCYR(NABCT)=((CYSD2-CCY(NABCT))/DELQRP)*2.0*V/NAC
DCLNR(NABCT)=((CLNSD2-CCLN(NABCT))/DELQRP)*2.0*V/SPAN
DCLLR(NABCT)=((CLLSD2-CCLL(NABCT))/DELQRP)*2.0*V/SPAN
IDFLGF=1
123
IDFLGB=1
GO TO 70
53
81
WRITE
50
WRITE(6,813)
GO TO 806
61
COMPUTED GO TO
IF THE VALUE OF IDSTAT IS 1 2
TRANSFER TO STATEMENT 62 63
62
CLSD1=CCL(NABCT)
CLNSD1=CCLN(NABCT)
CLLSD1=CCLL(NABCT)
CYSD1=CCY(NABCT)
CLNSD1=CCLN(NABCT)
CNSD1=CCN(NABCT)
IDSTAT=2
DELTAE=DELTS
GO TO 60
63
DCLD(NABCT)=(CLSD1-CCL(NABCT))/1.0
DCND(NABCT)=(CLNSD1-CCLN(NABCT))/1.0
DCLLD(NABCT)=(CLLSD1-CCLL(NABCT))/1.0
DCYD(NABCT)=(CYSD1-CCY(NABCT))/1.0
DCLNO(NABCT)=(CLNSD1-CCLN(NABCT))/1.0
DCNO(NABCT)=(CNSD1-CCN(NABCT))/1.0
IS
ABS(DCLLD(NABCT)).LT.1.0E-10
NO
YES
DCLLD(NABCT)=0.0

AERO

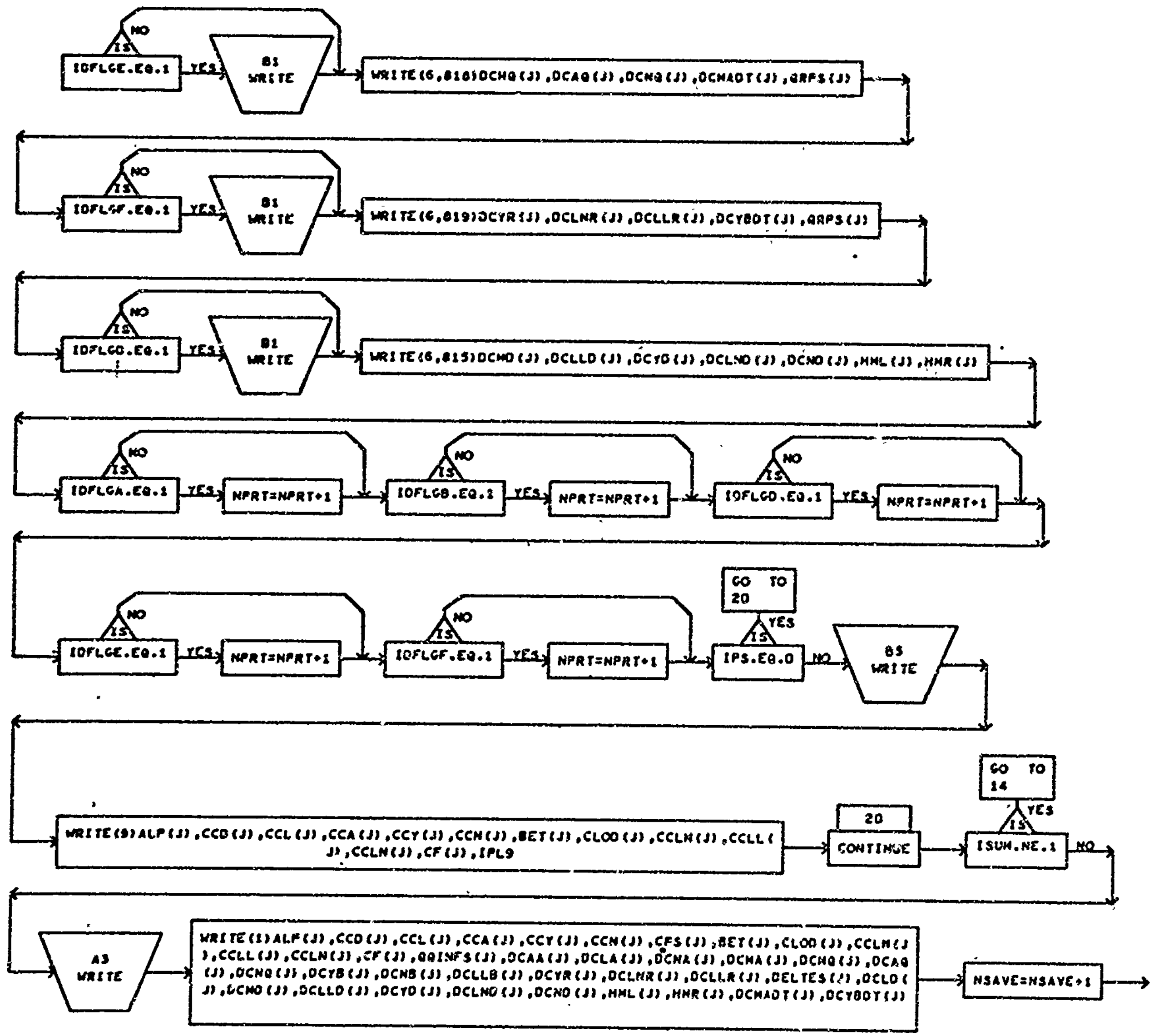


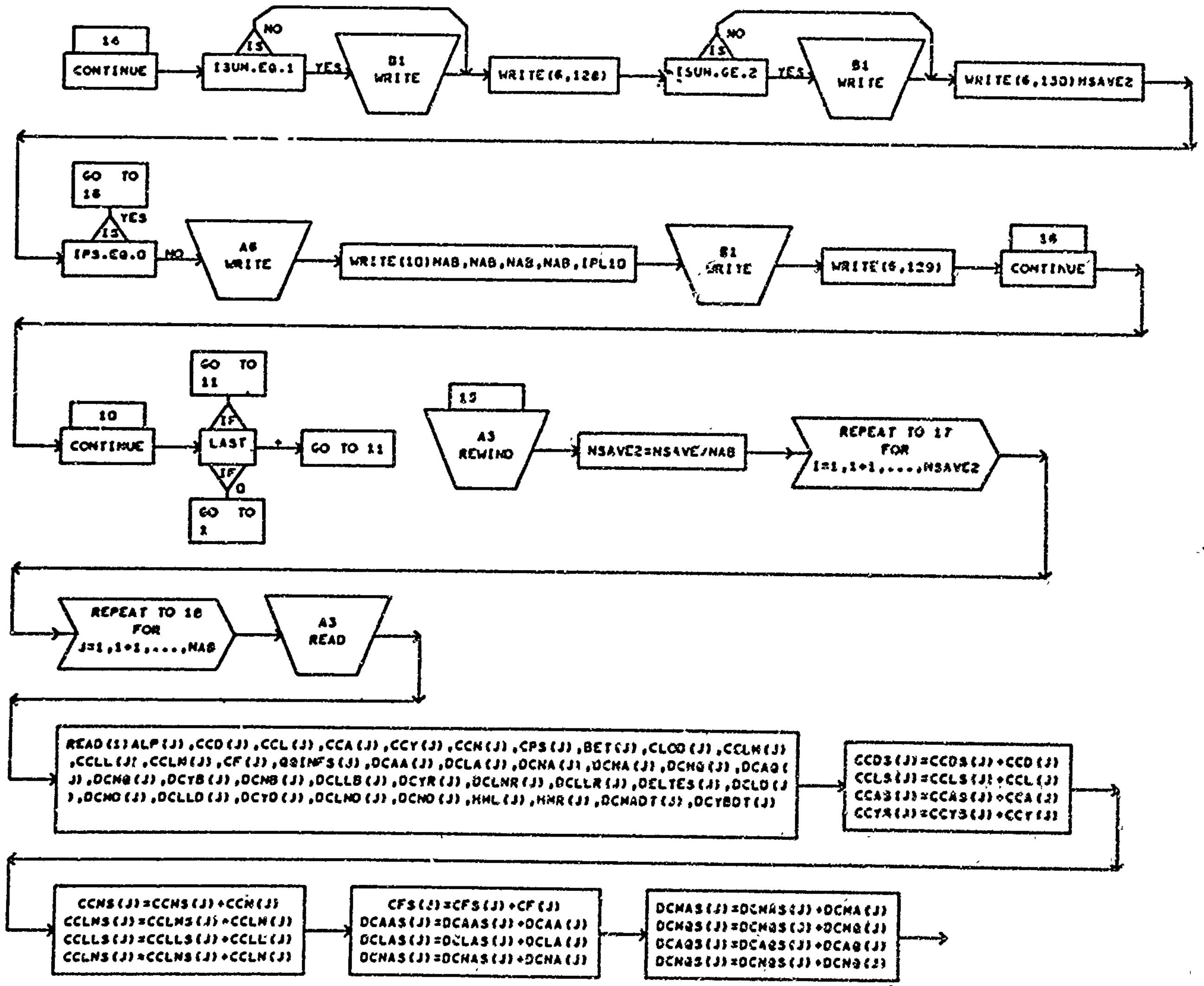
ABS(DCYD(NABCT)).LT.1.0E-10
IS
NO
YES
DCYD(NABCT)=0.0
ABS(DCLNO(NABCT)).LT.1.0E-10
DCLNO(NABCT)=0.0
IGFLGD=1
70
NABCT=NABCT+1
NABCT.LE.NAB
GO TO 2
21
CONTINUE
IABOOT.EQ.0
GO TO 125
REPEAT TO 124 FOR J=1,1+1,...,NAB
IDERIV.NE.5.AND.IDERIV.NE.6
GO TO 124
PERFORM SUBROUTINE PLUNGE
IDERS(NAB),DCHA(NAB),DCYB(NAB),DCHADT(NAB),DCYBDT(NAB)
ERROR.NE.0
GO TO 800
124
CONTINUE
125
CONTINUE
IVECT.EQ.0
GO TO 127
REPEAT TO 126 FOR J=1,1+1,...,NAB
PERFORM SUBROUTINE VECTOR
MACH,PFS,SREF,XCG,YCG,ZCG,SPAN,MAC,ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCN(J),BKT(J),CLOO(J),CCLM(J),CCLL(J),CCLN(J)
ERROR.NE.0
GO TO 800
126
CONTINUE
127
REPEAT TO 14 FOR J=1,1+1,...,NAB
NPRT.GE.28
GO TO 3
NPRT=NPRT+2
GO TO 4
3
NPRT=4
PERFORM SUBROUTINE HEADER

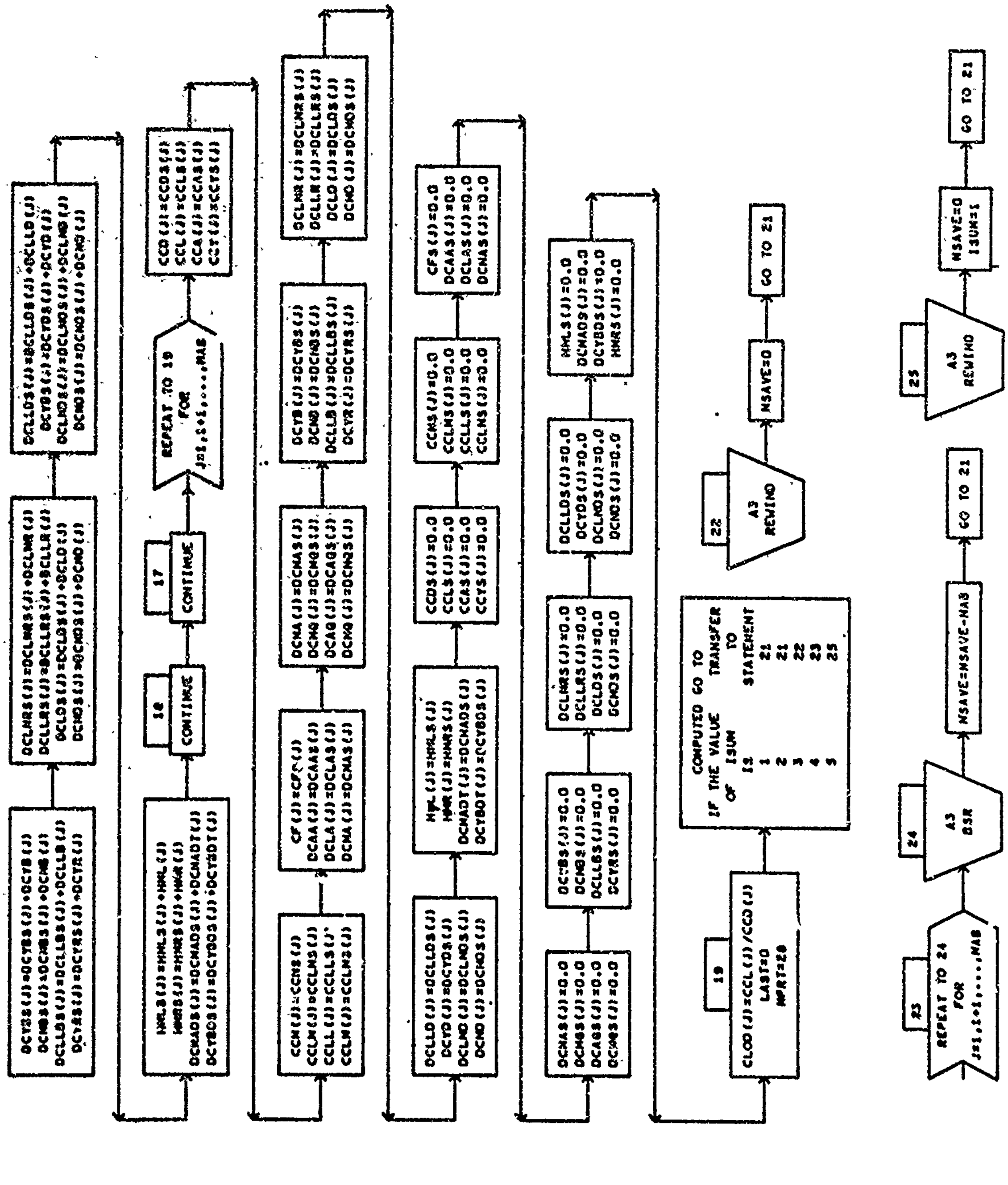

B1 WRITE
WRITE(6,105)MACH,V,RENO
IS PSTAG.LT.0.00001
YES
NO
B1 WRITE
WRITE(6,111)ALT
IS PSTAG.GT.0.00001
YES
NO
B1 WRITE
WRITE(6,112)PSTAG,TSTAG
B1 WRITE
WRITE(6,113)SREF,SPAN,MAC,XCG,YCG,ZCG
B1 WRITE
WRITE(6,114)
IS IDFLGA.EQ.1
YES
NO
B1 WRITE
WRITE(6,807)
IS IDFLGB.EQ.1
YES
NO
B1 WRITE
WRITE(6,808)
IS IDFLGE.EQ.1
YES
NO
B1 WRITE
WRITE(6,816)
IS IDFLGF.EQ.1
YES
NO
B1 WRITE
WRITE(6,817)
IS IDFLGD.EQ.1
YES
NO
B1 WRITE
WRITE(6,814)
4
B1 WRITE
4
WRITE(6,106)ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCM(J),CPS(J),IMP(J),ETAC3(J),IMPI(J),DELTES(J),BET(J),CLOO(J),CCLM(J),CCLL(J),CCLN(J),CF(J),QQINFS(J),ISH(J),ENPHS(J),ISHI(J)
IS IDFLGA.EQ.1
YES
NO
B1 WRITE
WRITE(6,809)DCAA(J),DCLA(J),DCNA(J),DCMA(J)
IS IDFLGB.EQ.1
YES
NO
B1 WRITE
WRITE(6,812)DCYB(J),DCNB(J),DCLLB(J)

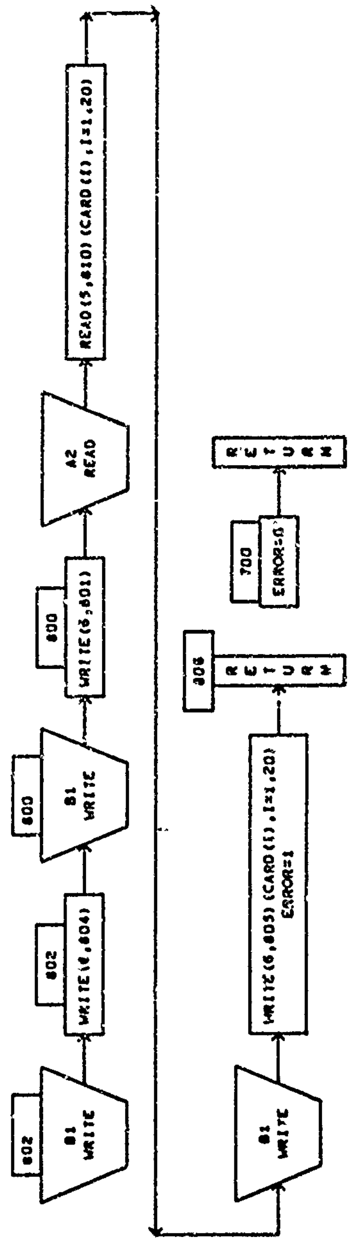
AERO

AERO


IDFLGE.EQ.1
IS
NO
YES
B1
WRITE
WRITE(6,818)DCHQ(J),DCAQ(J),DCNQ(J),DCHADT(J),QRPS(J)
IDFLGF.EQ.1
WRITE(6,819)DCYR(J),DCLNR(J),DCLLR(J),DCYBDT(J),QRPS(J)
IDFLGD.EQ.1
WRITE(6,815)DCHD(J),DCLLD(J),DCYD(J),DCLND(J),DCND(J),HML(J),HMR(J)
IDFLGA.EQ.1
NPRT=NPRT+1
IDFLGB.EQ.1
IDFLGD.EQ.1
IDFLGE.EQ.1
IDFLGF.EQ.1
GO TO 20
IPS.EQ.0
B5
WRITE
WRITE(9)ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCN(J),BET(J),CLOD(J),CCLM(J),CCLL(J),CCLN(J),CF(J),IPL9
20
CONTINUE
GO TO 14
ISUM.NE.1
A3
WRITE
WRITE(1)ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCN(J),CFS(J),BET(J),CLOD(J),CCLM(J),CCLL(J),CCLN(J),CF(J),QQINFS(J),DCAA(J),DCLA(J),DCNA(J),DCHA(J),DCHQ(J),DCAQ(J),DCNQ(J),DCYB(J),DCNB(J),DCLLB(J),DCYR(J),DCLNR(J),DCLLR(J),DELTES(J),DCLD(J),DCHD(J),DCLLD(J),DCYD(J),DCLND(J),DCND(J),HML(J),HMR(J),DCHADT(J),DCYBDT(J)
NSAVE=NSAVE+1


14
CONTINUE
ISUN.EQ.1
IS
NO
YES
B1
WRITE
WRITE(6,128)
ISUN.GE.2
IS
NO
YES
B1
WRITE
WRITE(6,130)NSAVE2
GO TO
18
YES
IS
IPS.EQ.0
NO
A6
WRITE
WRITE(10)NAB,NAB,NAB,NAB,IPL10
B1
WRITE
WRITE(6,129)
18
CONTINUE
10
CONTINUE
GO TO
11
IF
LAST
+
IF
0
GO TO
2
GO TO 11
15
A3
REWIND
NSAVE2=NSAVE/NAB
REPEAT TO 17
FOR
I=1,1+1,....,NSAVE2
REPEAT TO 18
FOR
J=1,1+1,....,NAB
A3
READ
READ(1)ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCN(J),CPS(J),BET(J),CLCD(J),CCLM(J)
,CCLL(J),CCLN(J),CF(J),QSINFS(J),DCAA(J),DCLA(J),DCNA(J),DCMA(J),DCMQ(J),DCAQ(
J),DCNQ(J),DCYB(J),DCNB(J),DCLLB(J),DCYR(J),DCLNR(J),DCLLR(J),DELTES(J),DCLD(J
),DCMD(J),DCLLD(J),DCYD(J),DCLND(J),DCMD(J),HML(J),HMR(J),DCMADT(J),DCYBDT(J)
CCDS(J)=CCDS(J)+CCD(J)
CCLS(J)=CCLS(J)+CCL(J)
CCAS(J)=CCAS(J)+CCA(J)
CCYS(J)=CCYS(J)+CCY(J)
CCNS(J)=CCNS(J)+CCN(J)
CCLMS(J)=CCLMS(J)+CCLM(J)
CCLLS(J)=CCLLS(J)+CCLL(J)
CCLNS(J)=CCLNS(J)+CCLN(J)
CFS(J)=CFS(J)+CF(J)
DCAAS(J)=DCAAS(J)+DCAA(J)
DCLAS(J)=DCLAS(J)+DCLA(J)
DCNAS(J)=DCNAS(J)+DCNA(J)
DCMAS(J)=DCMAS(J)+DCMA(J)
DCMQS(J)=DCMQS(J)+DCMQ(J)
DCAQS(J)=DCAQS(J)+DCAQ(J)
DCNQS(J)=DCNQS(J)+DCNQ(J)




802
B1
WRITE
802
WRITE(6,804)
800
B1
WRITE
800
WRITE(6,801)
A2
READ
READ(5,810)(CARD(I),I=1,20)
B1
WRITE
WRITE(6,805)(CARD(I),I=1,20)
ERROR=1
806
RETURN
700
ERROR=0
RETURN

SYMBOLS USED IN SUBROUTINE AERO

ALP	R	C	ANGLE OF ATTACK ARRAY	AERO
ALPHA	R	U	ANGLE OF ATTACK, DEGREES	AERO
ALPHAS	R	U	SAVED VALUE OF ANGLE OF ATTACK	AERO
ALT	R	U	ALTITUDE, FEET	AERO
AREA2	R	C	SURFACE AREA OF QUADRILATERALS	AERO
BET	R	C	YAW ANGLE ARRAY	AERO
BETA	R	U	YAW ANGLE, DEGREES	AERO
BETAS	R	U	SAVED VALUE OF YAW ANGLE	AERO
BS	R	C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	AERO
CAI	R	U	AXIAL FORCE INCREMENT	AERO
CARD	R	D	ARRAY FOR READING IN 80 COLUMN CARD	AERO
CASD1	R	U	FIRST VALUE OF AXIAL FORCE FOR DERIVATIVE CALCULATIONS	AERO
CASD2	R	U	SECOND VALUE OF AXIAL FORCE FOR DERIVATIVE CALCULATIONS	AERO
CASE	I	C	CASE NUMBER	AERO
CCA	R	C	AXIAL FORCE COEFFICIENT ARRAY	AERO
CCAS	R	D	SAVED VALUES OF AXIAL FORCE COEFFICIENT	AERO
CCD	R	C	DRAG COEFFICIENT ARRAY	AERO
CCDS	R	D	SAVED VALUES OF DRAG COEFFICIENT	AERO
CCL	R	C	LIFT COEFFICIENT ARRAY	AERO
CCLL	R	C	ROLLING MOMENT COEFFICIENT ARRAY	AERO
CCLLS	R	D	SAVED VALUES OF ROLLING MOMENT COEFFICIENT	AERO
CCLM	R	C	PITCHING MOMENT COEFFICIENT ARRAY	AERO
CCLMS	R	D	SAVED VALUES OF PITCHING MOMENT COEFFICIENT	AERO
CCLN	R	C	YAWING MOMENT COEFFICIENT ARRAY	AERO
CCLNS	R	D	SAVED VALUES OF YAWING MOMENT COEFFICIENT	AERO
CCLS	R	D	SAVED VALUES OF LIFT COEFFICIENT	AERO
CCN	R	C	NORMAL FORCE COEFFICIENT ARRAY	AERO
CCNS	R	D	SAVED VALUES OF NORMAL FORCE COEFFICIENT	AERO
CCY	R	C	SIDE FORCE COEFFICIENT ARRAY	AERO
CCYS	R	D	SAVED VALUES OF SIDE FORCE COEFFICIENT	AERO
CF	R	C	SKIN FRICTION TOTAL AXIAL FORCE CONTRIBUTION	AERO
CFS	R	D	SAVED VALUES OF SKIN FRICTION TOTAL AXIAL FORCE CONTRIBUTION	AERO
CLLI	R	U	ROLLING MOMENT COEFFICIENT INCREMENT	AERO
CLLSD1	R	U	FIRST VALUE OF ROLLING MOMENT FOR DERIVATIVE	AERO
CLLSD2	R	U	SECOND VALUE OF ROLLING MOMENT FOR DERIVATIVE	AERO
CLMI	R	U	PITCHING MOMENT COEFFICIENT INCREMENT	AERO

SYMBOLS USED IN SUBROUTINE AERO

CLMSD1	R	U	FIRST VALUE OF PITCHING MOMENT FOR DERIVATIVE	AERO
CLMSD2	R	U	SECOND VALUE OF PITCHING MOMENT FOR DERIVATIVE	AERO
CLNI	R	U	YAWING MOMENT COEFFICIENT INCREMENT	AERO
CLNSD1	R	U	FIRST VALUE OF YAWING MOMENT FOR DERIVATIVE	AERO
CLNSD2	R	U	SECOND VALUE OF YAWING MOMENT FOR DERIVATIVE	AERO
CLOD	R	C	LIFT TO DRAG RATIO ARRAY	AERO
CLSD1	R	U	FIRST VALUE OF LIFT COEFFICIENT FOR DERIVATIVE	AERO
CNI	R	U	NORMAL FORCE COEFFICIENT INCREMENT	AERO
CNSD1	R	U	FIRST VALUE OF NORMAL FORCE FOR DERIVATIVE	AERO
CNSD2	R	U	SECOND VALUE OF NORMAL FORCE FOR DERIVATIVE	AERO
CPS	R	C	ARRAY FOR NEWTONIAN CORRELATION FACTOR, K	AERO
CPSTAG	R	U	MODIFIED NEWTONIAN CORRELATION FACTOR, K	AERO
CYI	R	U	SIDE FORCE COEFFICIENT INCREMENT	AERO
CYSD1	R	U	FIRST VALUE OF SIDE FORCE COEFFICIENT FOR DERIVATIVE	AERO
CYSD2	R	U	SECOND VALUE OF SIDE FORCE COEFFICIENT FOR DERIVATIVE	AERO
DCAA	R	D	DERIVATIVE OF AXIAL FORCE WITH ANGLE OF ATTACK	AERO
DCAAS	R	D	SAVED VALUES OF AXIAL FORCE-ANGLE OF ATTACK DERIVATIVE	AERO
DCAQ	R	D	DERIVATIVE OF AXIAL FORCE WITH PITCH RATE	AERO
DCAQS	R	D	SAVED VALUES OF AXIAL FORCE DERIVATIVE WITH PITCH RATE	AERO
DCLA	R	D	DERIVATIVE OF LIFT COEFFICIENT WITH ANGLE OF ATTACK	AERO
DCLAS	R	D	SAVED LIFT COEFFICIENT DERIVATIVE WITH ANGLE OF ATTACK	AERO
DCLD	R	D	DERIVATIVE OF CL WITH CONTROL SURFACE DEFLECTION	AERO
DCLDS	R	D	SAVED VALUES OF CL DERIVATIVE WITH CONTROL DEFLECTION	AERO
DCLLB	R	D	DERIVATIVE OF ROLLING MOMENT WITH YAW ANGLE	AERO
DCLLBS	R	D	SAVED VALUES OF ROLLING MOMENT DERIVATIVE WITH YAW	AERO
DCLLD	R	D	DERIVATIVE OF ROLLING MOMENT WITH CONTROL DEFLECTION	AERO
DCLLDS	R	D	SAVED VALUES OF ROLLING MOMENT-CONTROL DERIVATIVE	AERO
DCLLR	R	D	DERIVATIVE OF ROLLING MOMENT WITH YAW RATE	AERO
DCLLRS	R	D	SAVED VALUES OF ROLLING MOMENT DERIVATIVE WITH YAW	AERO
DCLND	R	D	DERIVATIVE OF YAWING MOMENT WITH CONTROL DEFLECTION	AERO
DCLNDS	R	D	SAVED VALUES OF YAWING MOMENT-CONTROL DERIVATIVE	AERO
DCLNR	R	D	DERIVATIVE OF YAWING MOMENT WITH YAW RATE	AERO
DCLNRS	R	D	SAVED VALUES OF YAWING MOMENT-YAW RATE DERIVATIVE	AERO
DCMA	R	D	DERIVATIVE OF PITCHING MOMENT WITH ANGLE OF ATTACK	AERO
DCMADS	R	D	SAVED VALUES OF PITCHING MOMENT-ALPHA DOT DERIVATIVE	AERO
DCMADT	R	D	PITCHING MOMENT-ALPHA DOT DERIVATIVE	AERO

SYMBOLS USED IN SUBROUTINE AERO

DCMAS	R	D	SAVED VALUE OF PITCHING MOMENT-ALPHA DERIVATIVE	AERO
DCMD	R	D	PITCHING MOMENT-CONTROL DEFLECTION DERIVATIVE	AERO
DCMDS	R	D	SAVED VALUES OF PITCHING MOMENT-CONTROL DERIVATIVE	AERO
DCMQ	R	D	DERIVATIVE OF PITCHING MOMENT WITH PITCH RATE	AERO
DCMQS	R	D	SAVED VALUES OF PITCHING MOMENT-PITCH RATE DERIVATIVE	AERO
DCNA	R	D	DERIVATIVE OF NORMAL FORCE WITH ANGLE OF ATTACK	AERO
DCNAS	R	D	SAVED VALUE OF NORMAL FORCE-ALPHA DERIVATIVE	AERO
DCNB	R	D	DERIVATIVE OF YAWING MOMENT WITH YAW ANGLE	AERO
DCNBS	R	D	SAVED VALUE OF NORMAL FORCE-YAW DERIVATIVE	AERO
DCND	R	D	DERIVATIVE OF NORMAL FORCE WITH CONTROL DEFLECTION	AERO
DCNDS	R	D	SAVED VALUE OF NORMAL FORCE-CONTROL DERIVATIVE	AERO
DCNQ	R	D	DERIVATIVE OF NORMAL FORCE WITH PITCH RATE	AERO
DCNQS	R	D	SAVED VALUES OF NORMAL FORCE-PITCH RATE DERIVATIVES	AERO
DCYB	R	D	DERIVATIVE OF SIDE FORCE WITH YAW ANGLE	AERO
DCYBDS	R	D	SAVED VALUE OF CY-BETA DATA DERIVATIVE	AERO
DCYBDT	R	D	DERIVATIVE OF SIDE FORCE WITH BETA DOT	AERO
DCYBS	R	D	SAVED VALUE OF SIDE FORCE-YAW DERIVATIVE	AERO
DCYD	R	D	DERIVATIVE OF SIDE FORCE WITH CONTROL DEFLECTION	AERO
DCYDS	R	D	SAVED VALUES OF SIDE FORCE-CONTROL DERIVATIVE	AERO
DCYR	R	D	DERIVATIVE OF SIDE FORCE WITH YAW RATE	AERO
DCYRS	R	D	SAVED VALUES OF SIDE FORCE-YAW RATE DERIVATIVES	AERO
DELQRP	R	U	INCREMENT IN ROTATION RATE FOR ROTATION DERIVATIVES	AERO
DELTAE	R	U	CONTROL SURFACE DEFLECTION	AERO
DELTAS	R	U	SAVED VALUE OF CONTROL SURFACE DEFLECTION	AERO
DELTES	R	D	SAVED VALUE OF CONTROL SURFACE DEFLECTION	AERO
DELTS	R	U	SAVED VALUES OF CONTROL DEFLECTION	AERO
ENPM	R	U	SURFACE SLOPE MODIFICATION FACTOR	AERO
ENPMS	R	D	SAVED VALUES OF SURFACE SLOPE MODIFICATION FACTOR	AERO
ERROR	I	C	ERROR FLAG	AERO
ETAC	R	U	PRANDTL-MEYER-EXPANSION CORRECTION FACTOR	AERO
ETACS	R	D	SAVED VALUES OF PRANDTL-MEYER CORRECTION FACTOR	AERO
FS	R	C	FLOW PROPERTIES BEFORE SHOCK OR EXPANSION	AERO
HML	R	D	HINGE MOMENT (+Y SIDE OF VEHICLE)	AERO
HMLS	R	D	SAVED VALUES OF HINGE MOMENT (+Y)	AERO
HMLT	R	U	HINGE MOMENT (+Y)	AERO
HMR	R	D	HINGE MOMENT (-Y)	AERO

SYMBOLS USED IN SUBROUTINE AERO

HMRS	R	D	SAVED VALUES OF HINGE MOMENT (-Y)	AERO
HMRT	R	U	HINGE MOMENT (-Y)	AERO
I	I	U	DO-LOOP INDEX	AERO
IABDOT	I	U	ALPHA-DOT BETA-DOT DERIVATIVE FLAG	AERO
IDERIV	I	U	DERIVATIVE OPTION FLAG	AERO
IDERS	I	D	SAVED VALUES OF DERIVATIVE OPTION FLAG	AERO
IDFLGA	I	U	ALPHA DERIVATIVE PRINT FLAG	AERO
IDFLGB	I	U	BETA DERIVATIVE PRINT FLAG	AERO
IDFLGC	I	U	ROLL DERIVATIVE PRINT FLAG (NOT USED BY MARK II)	AERO
IDFLGD	I	U	CONTROL DERIVATIVE PRINT FLAG	AERO
IDFLGE	I	U	PITCH RATE DERIVATIVE PRINT FLAG	AERO
IDFLGF	I	U	YAW RATE DERIVATIVE PRINT FLAG	AERO
IDSTAT	I	U	DERIVATIVE CYCLE FLAG	AERO
IFIRST	I	U	FIRST POINT FLAG FOR USE IN NEWTPM	AERO
IFLG	I	U	DERIVATIVE FLAG	AERO
IGTYPE	I	U	COMPONENT TYPE (=1 FOR CONTROL)	AERO
IM	I	C	ELEMENT ROW NUMBER ARRAY	AERO
IMP	I	D	IMPACT METHOD ARRAY	AERO
IMPACI	I	U	STARTING ELEMENT IMPACT METHOD	AERO
IMPACT	I	U	IMPACT FORCE CALCULATION METHOD	AERO
IMPI	I	D	STARTING IMPACT METHOD ARRAY	AERO
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	AERO
IORIEN	I	U	ELEMENT ORIENTATION	AERO
IPL10	I	U	TYPE NUMBER FOR TAPE 10 DATA = 42	AERO
IPL9	I	U	TYPE NUMBER FOR TAPE 9 DATA =43	AERO
IPRINS	I	D	SAVED VALUES OF PRINT FLAG	AERO
IPRINT	I	U	PRINT FLAG FOR SHOCK-EXPANSION CALCULATIONS	AERO
IPRTS	I	D	SAVED VALUES OF IPRINT FLAG	AERO
IPS	I	U	SC-4020 AERO-DATA SAVE FLAG	AERO
IRET1	I	U	RETURN TO TYPE 1 CARD CONTROL FLAG	AERO
IREW8	I	U	REWIND TAPE 8 FLAG	AERO
IS	I	C	SKIN FRICTION CONTROL FLAG ARRAY	AERO
ISH	I	D	SHADOW METHOD ARRAY	AERO
ISHAD	I	U	SHADOW FORCE CALCULATION METHOD	AERO
ISHADI	I	U	STARTING ELEMENT METHOD IN SHADOW REGION	AERO
ISHI	I	D	SHADOW STARTING ELEMENT ARRAY	AERO

SYMBOLS USED IN SUBROUTINE AERO

ISIZ	I	U	NUMBER OF ELEMENTS TO BE STORED IN CORE	AERO
ISUM	I	U	COMPONENT SUMMATION AND SAVE FLAG	AERO
ITYP13	I	U	TYPE 13 CARD READ FLAG	AERO
IVECT	I	U	FORCE VECTOR METHOD FLAG	AERO
J	I	U	DO-LOOP INDEX	AERO
L	I	C	NUMBER OF ELEMENTS	AERO
LAST	I	U	LAST FLIGHT CONDITION FLAG	AERO
LS	I	C	NUMBER OF ELEMENTS	AERO
MAC	R	U	REFERENCE LENGTH FOR MOMENT COEFFICIENTS	AERO
MACH	R	U	MACH NUMBER	AERO
NAB	I	U	NUMBER OF ALPHA-BETA COMBINATIONS	AERO
NABCT	I	U	ALPHA-BETA COUNTER	AERO
NABS	I	U	SAVED VALUE FOR NUMBER OF ALPHA-BETA COMBINATIONS	AERO
NOAB	I	U	NO ALPHA-BETA CARD FLAG	AERO
NPRT	I	U	LINE COUNTER	AERO
NS	I	U	NUMBER OF SKIN FRICTION SURFACES	AERO
NSAVE	I	U	NUMBER OF SETS OF DATA SAVED FOR SUMMATION	AERO
NSAVE2	I	U	DO-LOOP INDEX FOR DATA SUMMATION	AERO
NW	R	U	WALL TEMPERATURE CALCULATION FLAG FOR FLOSEP	AERO
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	AERO
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	AERO
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	AERO
PAGE	I	C	PAGE NUMBER	AERO
PFS	R	U	FREE-STREAM PRESSURE-LBS/SQUARE FOOT	AERO
PRINT	I	U	DETAIL FORCE CONTRIBUTION PRINT FLAG	AERO
PRINTS	I	U	PRINT FLAG FOR ELEMENT DATA	AERO
PSTAG	R	U	WIND TUNNEL STAGNATION PRESSURE, ATMOSPHERES	AERO
QQINF	R	U	DYNAMIC PRESSURE RATIO CORRECTION FACTOR	AERO
QQINFS	R	C	SAVED VALUES OF DYNAMIC PRESSURE CORRECTION	AERO
QRP	R	U	INPUT VEHICLE ROTATION RATE, RADIANS/SECOND	AERO
QRPS	R	D	SAVED VALUES OF ROTATION RATE	AERO
QRPSS	R	U	SAVED VALUE OF INPUT ROTATION RATE	AERO
RENO	R	U	FREE STREAM REYNOLDS NUMBER	AERO
RETRAN	R	U	TRANSITION REYNOLDS NUMBER FOR CONTROL SURFACE	AERO
SPAN	R	U	REFERENCE LENGTH FOR ROLLING, YAWING COEFFICIENTS	AERO
SREF	R	U	VEHICLE REFERENCE AREA (WING AREA)	AERO

SYMBOLS USED IN SUBROUTINE AERO

SURF	R	C	SKIN FRICTION DATA ARRAY	AERO
SWEEP	R	U	LEADING EDGE SWEEP (NOT USED BY MARK II)	AERO
SYMFCT	I	U	SYMMETRY FLAG	AERO
TITLE	R	C	TITLE ARRAY	AERO
TSTAG	R	U	WIND TUNNEL STAGNATION TEMPERATURE-DEGREES F	AERO
TWALL	R	U	WALL TEMPERATURE FOR FLOSEP	AERO
TYPE	I	U	CARD TYPE	AERO
V	R	U	FREE-STREAM VELOCITY-FEET/SECOND	AERO
XCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-X	AERO
XCG	R	U	X-CENTER FOR MOMENT CALCULATIONS	AERO
YCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-Y	AERO
YCG	R	U	Y-CENTER FOR MOMENT CALCULATIONS	AERO
ZCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-Z	AERO
ZCG	R	U	Z-CENTER FOR MOMENT CALCULATIONS	AERO

3. SUBROUTINE HEADER (DECK AROB)

a. Algorithm

This routine provides the title at the top of each page of the output and advances the page counter.

b. Input/Output

Program header is printed at top of page on output Tape 6.

c. Error

None

d. Subroutines Required

None

e. Argument List

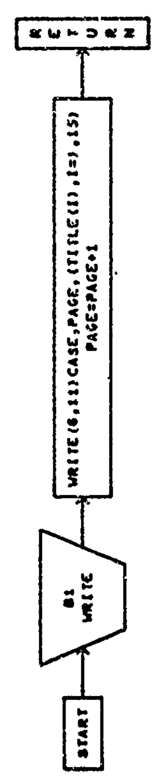
None

f. Length

342 bytes

```
DECK AROB

      SUBROUTINE HEADER                                                 AROB 0010
C                                                                       AROB 0020
      DIMENSION  TITLE(15)                                              AROB 0030
C                                                                       AROB 0040
      COMMON CASE,TITLE,PAGE                                            AROB 0050
C                                                                       AROB 0060
      INTEGER  PAGE, CASE                                               AROB 0070
C                                                                       AROB 0080
C  PRINT OUT HEADER AT TOP OF EACH PAGE OF OUTPUT                       AROB 0090
      WRITE (6,11) CASE,PAGE,(TITLE(I),I=1,15)                          AROB 0100
   11 FORMAT (1H1,49HHYPERSONIC ARBITRARY-BODY PROGRAM, MARK III MOD 0, AROB 0110
     1/,1H0,6H  CASE,I5,85X,5HPAGE I4,/1H0,14A4,1A3)                    AROB 0120
C                                                                       AROB 0130
C  STEP PAGE NUMBER BY ONF                                              AROB 0140
      PAGE = PAGE + 1                                                   AROB 0150
C                                                                       AROB 0160
      RETURN                                                            AROB 0170
      END                                                               AROB 0180
```

HEADER


START
81
WRITE
WRITE(6,11)CASE,PAGE,(TITLE(I),I=1,15)
PAGE=PAGE+1
RETURN

```
SYMBOLS USED IN SUBROUTINE HEADER

CASE     I  C  CASE NUMBER                                              HEADER
PAGE     I  C  PAGE NUMBER                                              HEADER
TITLE    R  C  TITLE                                                    HEADER
```

4. SUBROUTINE SDATA (DECK AROC)

This subroutine prepares geometry data for use by the rest of the program.

a. Algorithm

The Element Data Control Card (Type 2) is read and control passed to one of the other geometry routines (ANALY1, ANALY2, ANALY3) if required. The surface element data is then read (either from input Tape 5 or from the geometry storage Tape 8) and converted to quadrilaterals. These data are stored in core for the first 300 elements and on Tape 4 thereafter. All geometry data for control surface components are stored on Tapes 3 and 4.

b. Input/Output

Element Data Control Card (Type 2), Element Data Input Cards (Type 3).
When PRINTS is equal to 1 the Input Surface Element Data along with the direction cosines and centroid coordinates of each quadrilateral element are printed on output Tape 6.

c. Error

An error condition occurs when an input card type number is wrong.

d. Subroutines Required

ANALY1, ANALY2, ANALY3, HEADER

e. Argument List

(PRINTS, SYMFCT, ISIZ, IORIEN, IGTYPE)

f. Length

13264 bytes

DECK AROC

```
      SUBROUTINE SDATA (PRINTS,SYMFCT,ISIZ,IORIEN,IGTYPE)
C
C  THIS SUBROUTINE DETERMINES THE DIRECTION COSINES, CENTROID, AND AREA
C  OF THE INPUT SURFACE ELEMENTS.
C
      DIMENSION  XA(250),XB(250),YA(250),YB(250),ZA(250),ZB(250),
     1 XI(4),ETA(4),XIN(4),YIN(4),ZIN(4),TITLE(15),XPA(4),YPA(4),ZPA(4)
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),
     1  YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)
C
      COMMON  CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1  AREA2,IN,IM,L,LS
C
      REAL  NX,NY,NZ,LEFCT
      REAL  NX2,NY2,NZ2
      LOGICAL  RFLAG, AFLAG, BFLAG
      INTEGER  STAT, STATT, PAGE, CASE, TYPE, ERROR, PRINTS, SYMFCT, SEQ
    1 FORMAT (3F10.4,I1,3F10.4,I1,3X,I3,1A2,I2,4X,I4)
    2 FORMAT (3F10.4,I1,3F10.4,I1,3X,I3,1A2,I2,4HAERO,I4)
C
C     SET UP STARTING CONSTANTS
       N = -1
      NN = -1
       KLCT = 0
       NPRT = 10
       L = 0
       IGT = 0
       AREAT = 0.0
       VOL = 0.0
      WRITE (6,4016)
C  READ ELEMENT DATA CONTROL CARD
      READ (5,3) PRINTS,SYMFCT,IORIEN,IFACT,XSC,YSC,ZSC,DELX,DELY,DELZ,
     1  LEFCT,XLEO,IGEOM,ITAPE,IGTYPE,IELOV,TYPE
    3 FORMAT (I1,1XI1,2I1,1X3F6.0,1X3F6.0,1X,F5.0,F7.0,3X,4I1,7X,I2)
      IF (TYPE.EQ.2) GO TO 7
```

DECK AROC

```
      WRITE (6,5)
    5 FORMAT(1HC,46H***** ELEMENT DATA CONTROL CARD IS NOT PRESENT,
     1 35H OR HAS THE WRONG TYPE NUMBER *****  )
       GO TO 301
    7 IF (SYMFCT .EQ. 0) SYMFCT = 2
      IF((IGTYPE.EQ.1.OR.IGTYPE.EQ.3).AND.IORIEN.EQ.0)WRITE(6,290)IORIEN
  290 FORMAT (1H ,49H**** ON CONTROL SURFACE, ORIENTATION WAS INPUT AS
     1  2H =I2,49H  PROGRAM CONTINUED WITH ORIENTATION SET = 1 ****  )
      IF ((IGTYPE.EQ.1.OR.IGTYPE.EQ.3).AND.IORIEN.EQ.0) IORIEN = 1
      IA = 0
      IB = 0
      IF (IORIEN .EQ. 2) IB = 1
      IF (IORIEN .EQ. 3) IA = 1
      IF (IORIEN .GT. 1) IORIEN = 1
      IF (IGTYPE .GT. 0) IGT = 1
      IF (IGEOM .EQ. 0) GO TO 4
C  GEOMETRY IS TO BE CALCULATED BY ONE OF THE ANALYTICAL SHAPE ROUTINES
      GO TO (11,12,13),IGEOM
   11 CALL ANALY1
      GO TO 4
   12 CALL ANALY2
      GO TO 4
   13 CALL ANALY3
    4 IF (ERROR .NE. 0) GO TO 6
      IF (ITAPE.EQ.1 .OR. ITAPE.EQ.3) REWIND 8
      IF (ITAPE .EQ. 5) GO TO 6
C
C     READ IN ALL SURFACE DATA
   29 IF (ITAPE.EQ.0 .OR. ITAPE.EQ.3 .OR. ITAPE.EQ.4) READ (5,1) X,Y,Z,
     1    STAT, XX,YY,ZZ, STATT, CASE, SECT, TYPE, SEQ
      IF (ITAPE.EQ.1 .OR. ITAPE.EQ.2) READ (8,1) X,Y,Z,STAT, XX,YY,ZZ,
     1     STATT, CASE, SECT, TYPE, SEQ
      STAT =IABS(STAT)
      STATT =IABS(STATT)
      IF (ITAPE.EQ.3 .OR. ITAPE.EQ.4) WRITE (8,2) X,Y,Z,STAT, XX,YY,ZZ,
     1     STATT, CASE, SECT, TYPE, SEQ
```

```
DECK AROC

      IF (TYPE.NE.3) GO TO 300                                          AROC  0720
      IF ((STAT.EQ.0.OR.STATT.EQ.0).AND.(STAT.NE.2.AND.STATT.NE.2))     AROC  0730
     1      SECTS = SECT                                                AROC  0740
      RFLAG = .FALSE.                                                   AROC  0750
      GO TO 80                                                          AROC  0760
   30 IF (RFLAG) GO TO 50                                               AROC  0770
      RFLAG = .TRUE.                                                    AROC  0780
      X = XX                                                            AROC  0790
      Y = YY                                                            AROC  0800
      Z = ZZ                                                            AROC  0810
      STAT = STATT                                                      AROC  0820
      GO TO 60                                                          AROC  0830
   50 RFLAG = .FALSE.                                                   AROC  0840
      IF (ITAPE.EQ.0 .OR. ITAPE.EQ.3 .OR. ITAPE.EQ.4) READ (5,1) X,Y,Z, AROC  0850
     1    STAT, XX,YY,ZZ, STATT, CASE, SECT, TYPE, SEQ                  AROC  0860
      IF (ITAPE.EQ.1 .OR. ITAPE.EQ.2) READ (8,1) X,Y,Z,STAT, XX,YY,ZZ,  AROC  0870
     1     STATT, CASE, SECT, TYPE, SEQ                                 AROC  0880
      STAT =IABS(STAT)                                                  AROC  0890
      STATT =IABS(STATT)                                                AROC  0900
      IF (ITAPE.EQ.3 .OR. ITAPE.EQ.4) WRITE (8,2) X,Y,Z,STAT, XX,YY,ZZ, AROC  0910
     1     STATT, CASE, SECT, TYPE, SEQ                                 AROC  0920
      IF (TYPE.NE.3) GO TO 300                                          AROC  0930
      IF ((STAT.EQ.0.OR.STATT.EQ.0).AND.(STAT.NE.2.AND.STATT.NE.2))     AROC  0940
     1      SECTS = SECT                                                AROC  0950
   60 IF (STAT .EQ. 0 .OR. STAT .EQ. 3) GO TO 180                       AROC  0960
      IF (STAT .EQ. 2) GO TO 200                                        AROC  0970
   70 IF (.NOT. AFLAG) GO TO 200                                        AROC  0980
      MC = M                                                            AROC  0990
   80 M = 1                                                             AROC  1000
       IF (STAT .EQ. 2) GO TO 150                                       AROC  1010
       IF (.NOT. BFLAG) GO TO 84                                        AROC  1020
   75 DO 81 J =1,MC                                                     AROC  1030
        XA(J) = XB(J)                                                   AROC  1040
        YA(J) = YB(J)                                                   AROC  1050
   81   ZA(J) = ZB(J)                                                   AROC  1060
   83 XB(1) = X                                                         AROC  1070
```

DECK AROC

```
      YB(1) = Y                                                         AROC  1080
      ZB(1) = Z                                                         AROC  1090
      GO TO 30                                                          AROC  1100
   84 IF (AFLAG) GO TO 85                                               AROC  1110
       BFLAG = .TRUE.                                                   AROC  1120
       GO TO 75                                                         AROC  1130
   85 AFLAG = .FALSE.                                                   AROC  1140
       GO TO 83                                                         AROC  1150
  150 AFLAG = .TRUE.                                                    AROC  1160
       BFLAG = .FALSE.                                                  AROC  1170
       N = N+1                                                          AROC  1180
      NN = NN + 1                                                       AROC  1190
  160 XA(M) = X                                                         AROC  1200
      YA(M) = Y                                                         AROC  1210
      ZA(M) = Z                                                         AROC  1220
      GO TO 30                                                          AROC  1230
  180 M = M + 1                                                         AROC  1240
      IF (AFLAG) GO TO 160                                              AROC  1250
       XB(M) = X                                                        AROC  1260
       YB(M) = Y                                                        AROC  1270
       ZB(M) = Z                                                        AROC  1280
      IF (STAT .NE. 3) GO TO 30                                         AROC  1290
  200 MMIN = MINO (M,MC) - 1                                            AROC  1300
       NN2 = 1                                                          AROC  1310
       MC = M                                                           AROC  1320
  250 N = N + 1                                                         AROC  1330
      NN = NN + 1                                                       AROC  1340
       KLCT = KLCT + 1                                                  AROC  1350
       JJ = 0                                                           AROC  1360
C                                                                       AROC  1370
C  BEGIN COMPUTATION OF SURFACE ELEMENT CHARACTERISTICS                 AROC  1380
  450 IF (IELOV .EQ. 1) GO TO 2001                                      AROC  1390
      DO 2000 I= 1,MMIN                                                 AROC  1400
      IIA = I + IA                                                      AROC  1410
      IIB = I + IB                                                      AROC  1420
      IF (IFACT.EQ.1) GO TO 460                                         AROC  1430
```

```
      XIN(1) = XA(IIA )                           AROC
      XIN(2) = XA(IIA +1)                         AROC
      XIN(3) = XB(IIB +1)                         AROC
      XIN(4) = XB(IIB )                           AROC
      YIN(1) = YA(IIA )                           AROC
      YIN(2) = YA(IIA +1)                         AROC
      YIN(3) = YB(IIB +1)                         AROC
      YIN(4) = YB(IIB )                           AROC
      ZIN(1) = ZA(IIA )                           AROC
      ZIN(2) = ZA(IIA +1)                         AROC
      ZIN(3) = ZB(IIB +1)                         AROC
      ZIN(4) = ZB(IIB )                           AROC
     GO TO 201                                    AROC
C                                                 AROC
 460  XIN(1) = XA(IIA )   * XSC + DELX            AROC
      XIN(2) = XA(IIA +1) * XSC + DELX            AROC
      XIN(3) = XB(IIB +1) * XSC + DELX            AROC
      XIN(4) = XB(IIB )   * XSC + DELX            AROC
      YIN(1) = YA(IIA )   * YSC + DELY            AROC
      YIN(2) = YA(IIA +1) * YSC + DELY            AROC
      YIN(3) = YB(IIB +1) * YSC + DELY            AROC
      YIN(4) = YB(IIB )   * YSC + DELY            AROC
      ZIN(1) = ZA(IIA )   * ZSC + DELZ            AROC
      ZIN(2) = ZA(IIA +1) * ZSC + DELZ            AROC
      ZIN(3) = ZB(IIB +1) * ZSC + DELZ            AROC
      ZIN(4) = ZB(IIB )   * ZSC + DELZ            AROC
C  FORM DIAGONAL VECTORS                          AROC
 201  T1X = XIN(3) - XIN(1)                       AROC
      T2X = XIN(4) - XIN(2)                       AROC
      T1Y = YIN(3) - YIN(1)                       AROC
      T2Y = YIN(4) - YIN(2)                       AROC
      T1Z = ZIN(3) - ZIN(1)                       AROC
      T2Z = ZIN(4) - ZIN(2)                       AROC
C                                                 AROC
C  FORM CROSS PRODUCT   N=T2 X T1                 AROC
      NX = T2Y*T1Z - T1Y*T2Z                      AROC
```

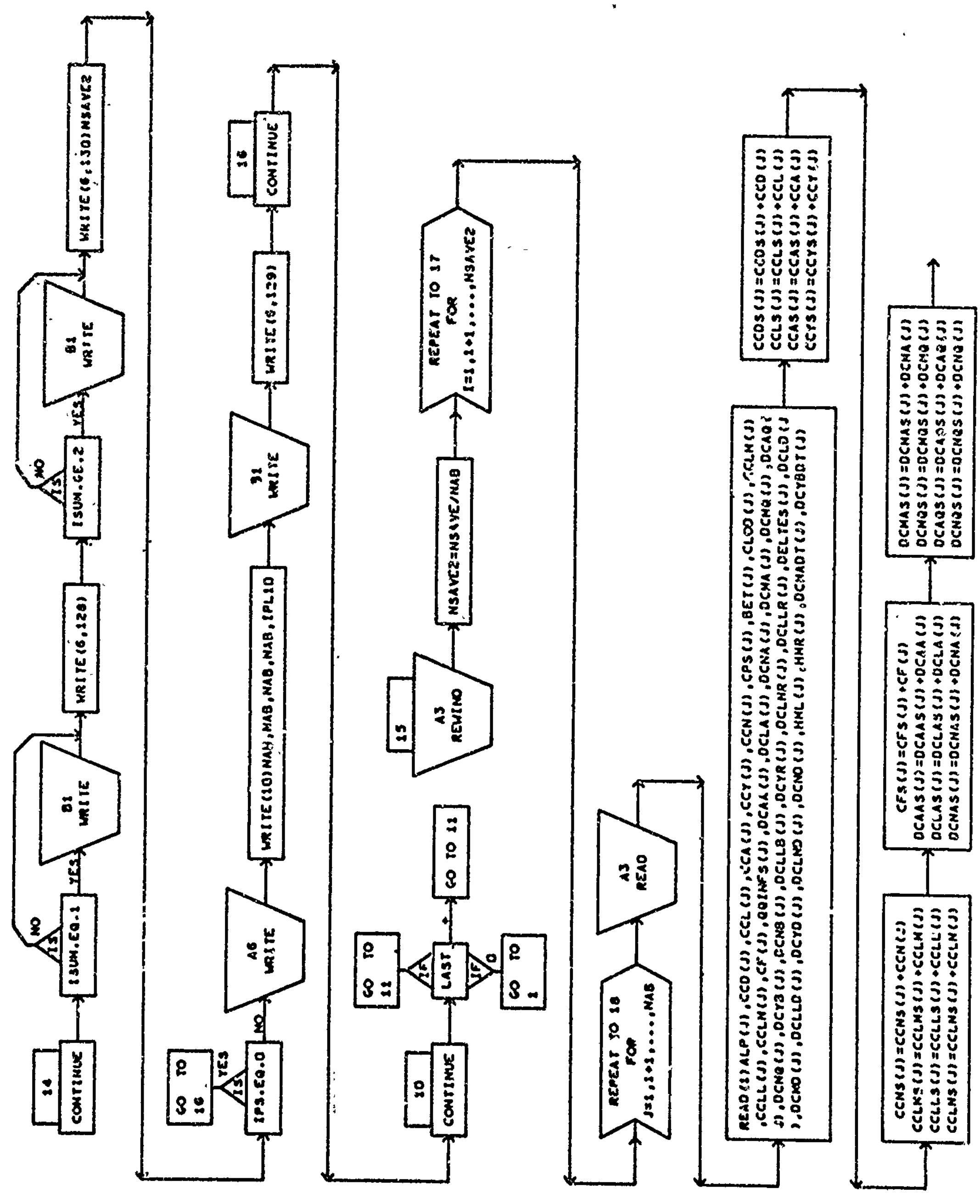
DECK AROC

```
      NY = T1X*T2Z - T2X*T1Z
      NZ = T2X*T1Y - T1X*T2Y
      VN = SQRT ( NX*NX + NY*NY + NZ*NZ )
     IF (VN .EQ. 0.0) GO TO 601
C
C  FORM UNIT NORMAL VECTOR
      NX = NX / VN
      NY = NY / VN
      NZ = NZ / VN
C
C  COMPUTE AVERAGE POINT
 601  AVX = 0.25 * (XIN(1) + XIN(2) + XIN(3) +  XIN(4) )
      AVY = 0.25 * (YIN(1) + YIN(2) + YIN(3) + YIN(4) )
      AVZ = 0.25 * (ZIN(1) + ZIN(2) + ZIN(3) + ZIN(4) )
C
C  COMPUTE PROJECTION DISTANCE
      D = NX*(AVX - XIN(1)) + NY*(AVY - YIN(1)) + NZ*(AVZ-ZIN(1))
      PD = ABS(D)
C
      T = SQRT (T1X*T1X + T1Y*T1Y + T1Z*T1Z)
     IF (T .EQ. 0.0) GO TO 603
     T1X = T1X / T
      T1Y = T1Y / T
      T1Z = T1Z / T
C
 603  T2X = NY*T1Z - NZ*T1Y
      T2Y = NZ*T1X - NX*T1Z
      T2Z = NX*T1Y - NY*T1X
C
C  COMPUTE COORDINATES OF CORNER POINTS IN REFERENCE COORD. SYSTEM
     DO 1000 J = 1,4
      XPA(J) = XIN(J) + NX*D
      YPA(J) = YIN(J) + NY*D
      ZPA(J) = ZIN(J) + NZ*D
      D = - D
      XDIF = XPA(J) - AVX
```

DECK AROC

```
      YDIF = YPA(J) - AVY
      ZDIF = ZPA(J) - AVZ
C
C  TRANSFORM CORNER POINTS TO ELEMENT COORDINATE SYSTEM (XI,ETA) WITH
C   AVERAGE POINT AS ORIGIN
      XI(J) = T1X*XDIF + T1Y*YDIF + T1Z*ZDIF
 1000 ETA(J)= T2X*XDIF + T2Y*YDIF + T2Z*ZDIF
     ETACK = ETA(2) - ETA(4)
     IF (ETACK .NE.0.0) GO TO 312
      XI0 = 0.0
      GO TO 313
C
C  COMPUTE CENTROID
 312 XI0 = .333333333 * (XI(4) * (ETA(1)-ETA(2)) + XI(2)
    1       * (ETA(4)-ETA(1))) / (ETA(2)-ETA(4))
 313 ETA0 = -.333333333 * ETA(1)
C
C  OBTAIN CORNER POINTS IN SYSTEM WITH CENTROID AS ORIGIN
      DO 1020 J = 1,4
       XI(J) = XI(J) - XI0
 1020  ETA(J) = ETA(J) - ETA0
C
C  TRANSFORM CENTROID TO REFERENCE COORDINATE SYSTEM
      XCENT = AVX + T1X*XI0 + T2X*ETA0
      YCENT = AVY + T1Y*XI0 + T2Y*ETA0
      ZCENT = AVZ + T1Z*XI0 + T2Z*ETA0
C
C  CONSTANTS FOR USE IN COMPUTING AREA OF ELEMENT
      XI3M1 = XI(3) - XI(1)
      ETA2M4 = ETA(2) - ETA(4)
C
C  COMPUTE AREA AND VOLUME OF ELEMENTS
      AREA = 0.5 * XI3M1 * ETA2M4
     AREAT = AREAT + AREA
      DELVOL = AREA * NY * YCENT
     VOL = VOL + DELVOL
```

```
DECK AROC
      L = L + 1                                                         AROC  2520
      II = I                                                            AROC  2530
      IF (PRINTS.EQ.0) GO TO 1770                                       AROC  2540
C                                                                       AROC  2550
C                                                                       AROC  2560
C   PRINT RESULTS OF CALCULATICNS TO DETERMINE ELEMENT  CHARACTERISTICS AROC  2570
 1700 IF (NPRT .GE.9)   GO TO 1750                                      AROC  2580
       NPRT = NPRT + 1                                                  AROC  2590
      IF  (I .EQ. 1) GO TO 1760                                         AROC  2600
       WRITE (6,4005) I, XIN, NX, XCENT, AREA,L,YIN,NY,YCENT,DELVOL,ZIN, AROC  2610
     1  NZ,ZCENT,VOL                                                    AROC  2620
      GO TO 1770                                                        AROC  2630
 1750 NPRT = 0                                                          AROC  2640
      CALL HEADER                                                       AROC  2650
      WRITE (6, 4002)                                                   AROC  2660
 1760 WRITE (6, 4010) N, I, XIN, NX, XCENT, AREA,L,YIN,NY,YCENT,DELVOL, AROC  2670
     1 ZIN,NZ,ZCENT,VOL                                                 AROC  2680
C                                                                       AROC  2690
C   SET UP DATA TO BE SAVED AND USED IN FORCE CALCULATIONS              AROC  2700
 1770 IF (IGT .GT. 0) GO TO 1772                                        AROC  2710
      IF (L .LE. ISIZ) GO TO 1771                                       AROC  2720
      WRITE (4) L,N,I,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA                   AROC  2730
      GO TO 2000                                                        AROC  2740
 1771  NX2(L) = NX                                                      AROC  2750
       NY2(L) = NY                                                      AROC  2760
       NZ2(L) = NZ                                                      AROC  2770
       XCENT2(L) = XCENT                                                AROC  2780
       YCENT2(L) = YCENT                                                AROC  2790
       ZCENT2(L)  = ZCENT                                               AROC  2800
       AREA2(L)  = AREA                                                 AROC  2810
       IN(L)     = N                                                    AROC  2820
       IM(L)     = I                                                    AROC  2830
      GO TO 2000                                                        AROC  2840
C   SAVE GEOMETRY DATA FOR CONTROL SURFACE USE                          AROC  2850
 1772  IF (L.GT. 1) GO TO 1600                                          AROC  2860
      XLEP1 = XPA(1)                                                    AROC  2870
```



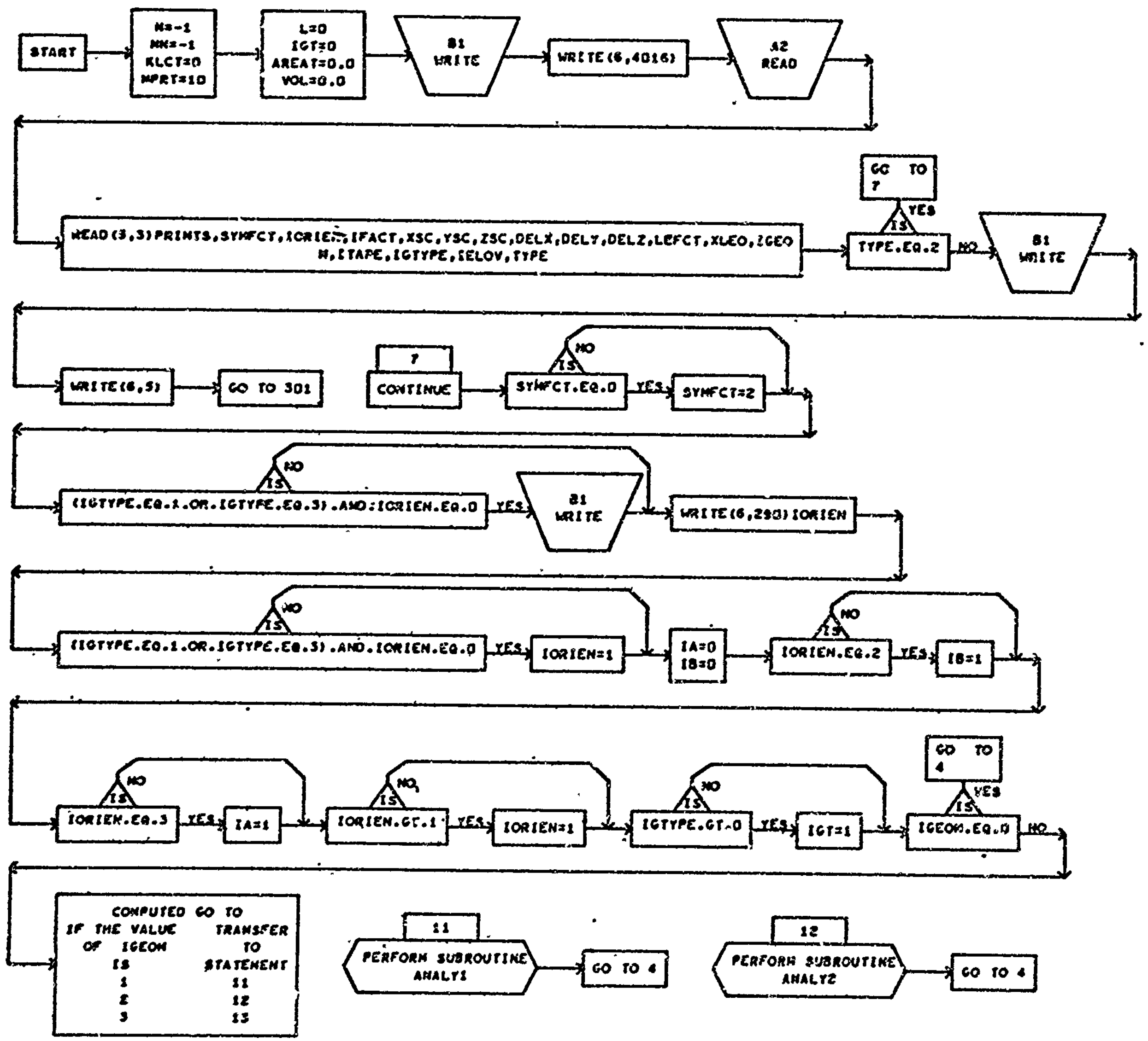
14
CONTINUE
IS
ISUM.EQ.1
NO
YES
B1
WRITE
WRITE(6,128)
IS
ISUM.GE.2
NO
YES
B1
WRITE
WRITE(6,130)NSAVE2
GO TO 16
IS
IPS.EQ.0
YES
NO
A6
WRITE
WRITE(10)NAH,NAB,NAB,NAB,IPLID
B1
WRITE
WRITE(6,129)
16
CONTINUE
10
CONTINUE
GO TO 11
IF
LAST
IF
0
GO TO 1
GO TO 11
15
A3
REWIND
NSAVE2=NSAVE/NAB
REPEAT TO 17
FOR
I=1,1+1,...,NSAVE2
REPEAT TO 18
FOR
J=1,1+1,...,NAB
A3
READ
READ(8)ALP(J),CCD(J),CCL(J),CCA(J),CCY(J),CCN(J),CPS(J),BET(J),CLOO(J),CCLM(J)
,CCLL(J),CCLN(J),CF(J),QQINFS(J),DCAA(J),DCLA(J),DCNA(J),DCNA(J),DCMQ(J),DCAQ(
J),DCNQ(J),DCY3(J),CCNB(J),DCLLB(J),DCYR(J),DCLNR(J),DCLLR(J),DELTES(J),DCLD(J
),DCMO(J),DCLLD(J),DCYD(J),DCLND(J),DCNO(J),HML(J),HMR(J),DCNADT(J),DCYBDT(J)
CCDS(J)=CCDS(J)+CCD(J)
CCLS(J)=CCLS(J)+CCL(J)
CCAS(J)=CCAS(J)+CCA(J)
CCYS(J)=CCYS(J)+CCY(J)
CCNS(J)=CCNS(J)+CCN(J)
CCLMS(J)=CCLMS(J)+CCLM(J)
CCLLS(J)=CCLLS(J)+CCLL(J)
CCLNS(J)=CCLNS(J)+CCLN(J)
CFS(J)=CFS(J)+CF(J)
DCAAS(J)=DCAAS(J)+DCAA(J)
DCLAS(J)=DCLAS(J)+DCLA(J)
DCNAS(J)=DCNAS(J)+DCNA(J)
DCMAS(J)=DCMAS(J)+DCMA(J)
DCMQS(J)=DCMQS(J)+DCMQ(J)
DCAQS(J)=DCAQS(J)+DCAQ(J)
DCNQS(J)=DCNQS(J)+DCNQ(J)

DECK AROC

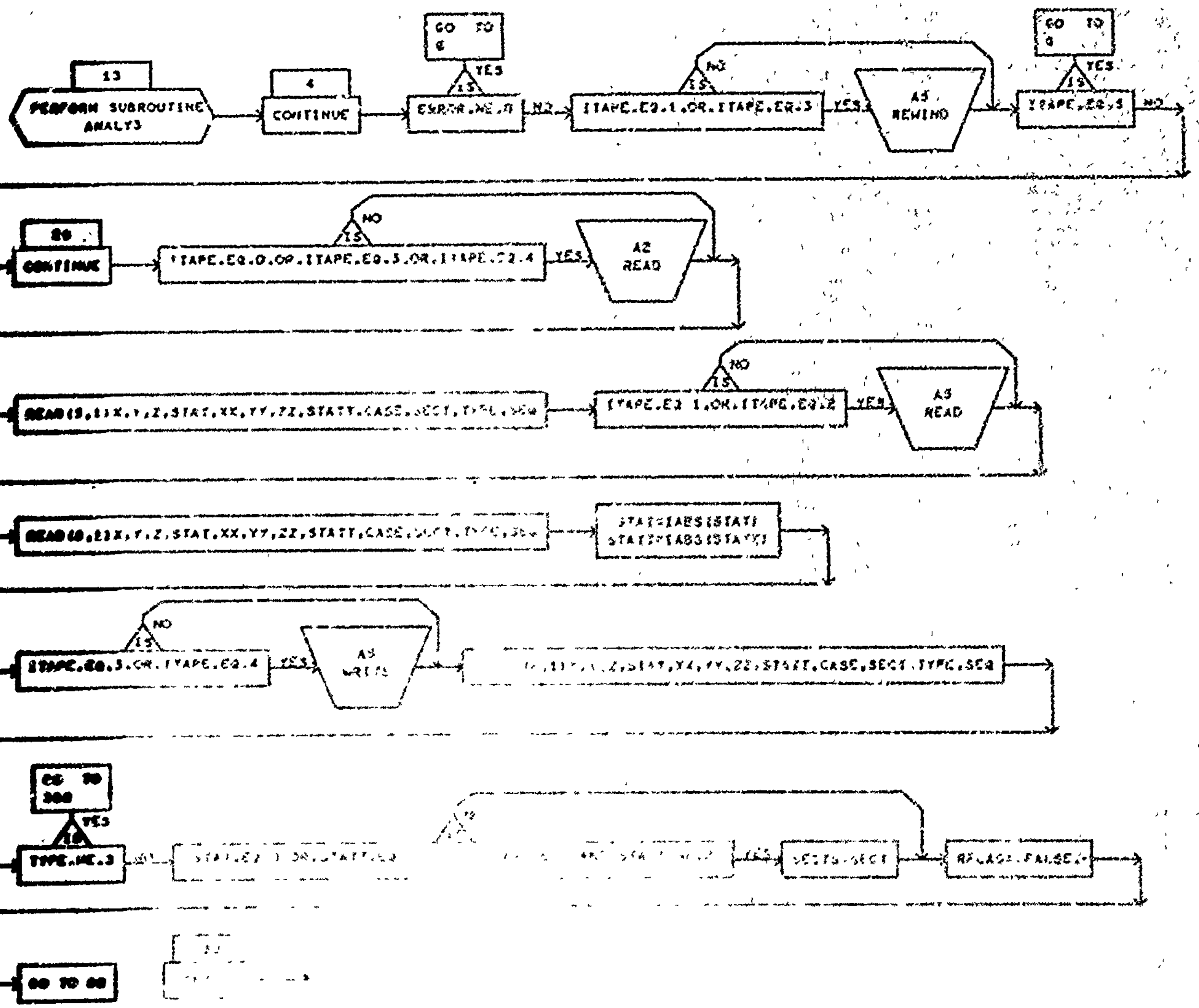
```
      IF (IGT .FQ. 2) GO TO 1650
      NX2(2) = XIN(3)
      NY2(2) = YIN(3)
      NZ2(2) = ZIN(3)
      IN(1) = L
      NFS = N
      IA = 0
      IB = 0
      IGT = 2
      GO TO 2020
 1650 IN(2) = L - IN(1)
      IN(4) = II
      IF (N .EQ. NFS) GO TO 2020
      WRITE (6,1651)
 1651 FORMAT (1H ,49H***** NUMBER OF STREAMWISE STRIPS ON FORE-SURFACE
     1  55H AND FLAP MUST BE THE SAME.  CHANGE GEOMETRY DATA ***** )
      ERROR = 3
      GO TO 6
C
C  TEST FOR END OF CASE
 2020 IF (STAT .NE. 3) GO TO 80
      GO TO 302
C
C  ERROR CHECK ON READING CARDS
 300  WRITE (6,4003)
C
 4003 FORMAT (1H0,50H***** SURFACE DATA ROUTINE HAS ATTEMPYED TO READ A
     142H NON SURFACE CARD - CHECK YOUR CARDS *****  )
 301  ERROR = 1
      RETURN
C
 302  LS = L
      WRITE (6,4016)
 4016 FORMAT (1H1 )
      REWIND 3
      REWIND 4
```

```
DECK AROC

    6 RETURN                                                            AROC 3600
C                                                                       AROC 3610
 4005 FORMAT (1H0,7X, I4, 1P4E14.5,0PF10.6,1P2E14.5,I6,2(/12X,4E14.5,   AROC 3620
     1 0PF10.6,1P2E14.5)  )                                             AROC 3630
C                                                                       AROC 3640
 4010 FORMAT (1H0,3X, 2I4,1P4E14.5,0PF10.6,1P2E14.5,I6,2(/12X,4E14.5,   AROC 3650
     1 0PF10.6,1P2E14.5)  )                                             AROC 3660
C                                                                       AROC 3670
 4015 FORMAT (1H0,10H SECTION =1A2,33H   TOTAL AREA OF INPUT ELEMENTS = AROC 3680
     1 F12.3,6X26HTOTAL NUMBER OF ELEMENTS = I5/1H ,12X,                AROC 3690
     2 33H TOTAL VOLUME OF INPUT ELEMENTS =F12.3,/1H0,3(20X,            AROC 3700
     3 9H*********))                                                    AROC 3710
C                                                                       AROC 3720
 4002 FORMAT (1H0,28H  INPUT SURFACE ELEMENT DATA/1H0,6X1HN3X1HM7X1HX,  AROC 3730
     1 3(13X,1HX),11X2HNX9X5HXCENT9X4HAREA8X1HL ,/1H ,5X, 4(13X,1HY),   AROC 3740
     2 11X2HNY9X5HYCENT ,7X,7HDELTA V,/1H ,5X,4(13X,1HZ),11X2HNZ,       AROC 3750
     3 9X,5HZCENT ,7X,6HVOLUME,/1H  )                                   AROC 3760
C                                                                       AROC 3770
      END                                                               AROC 3780
```

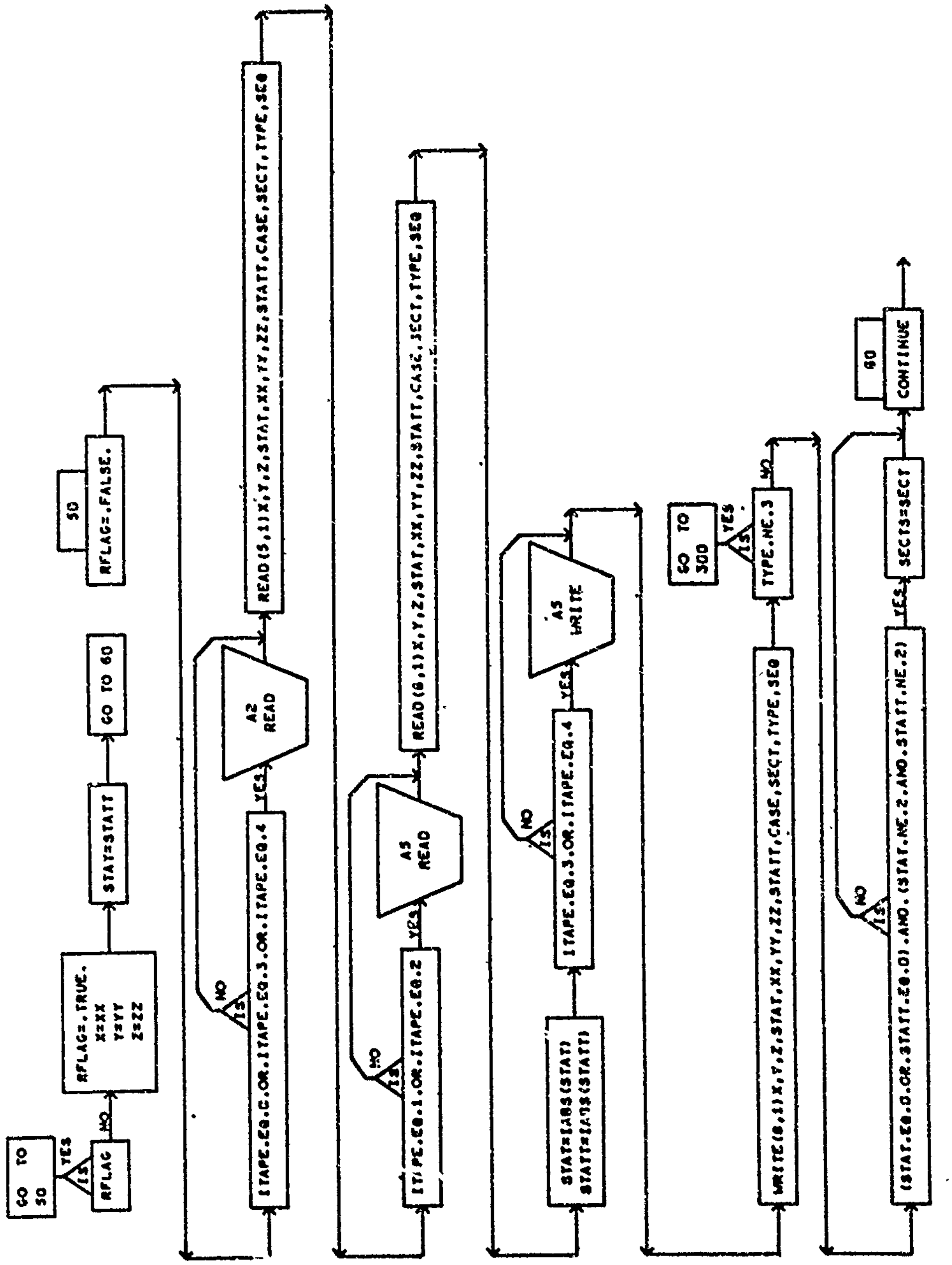
SUBROUTINE SDATA



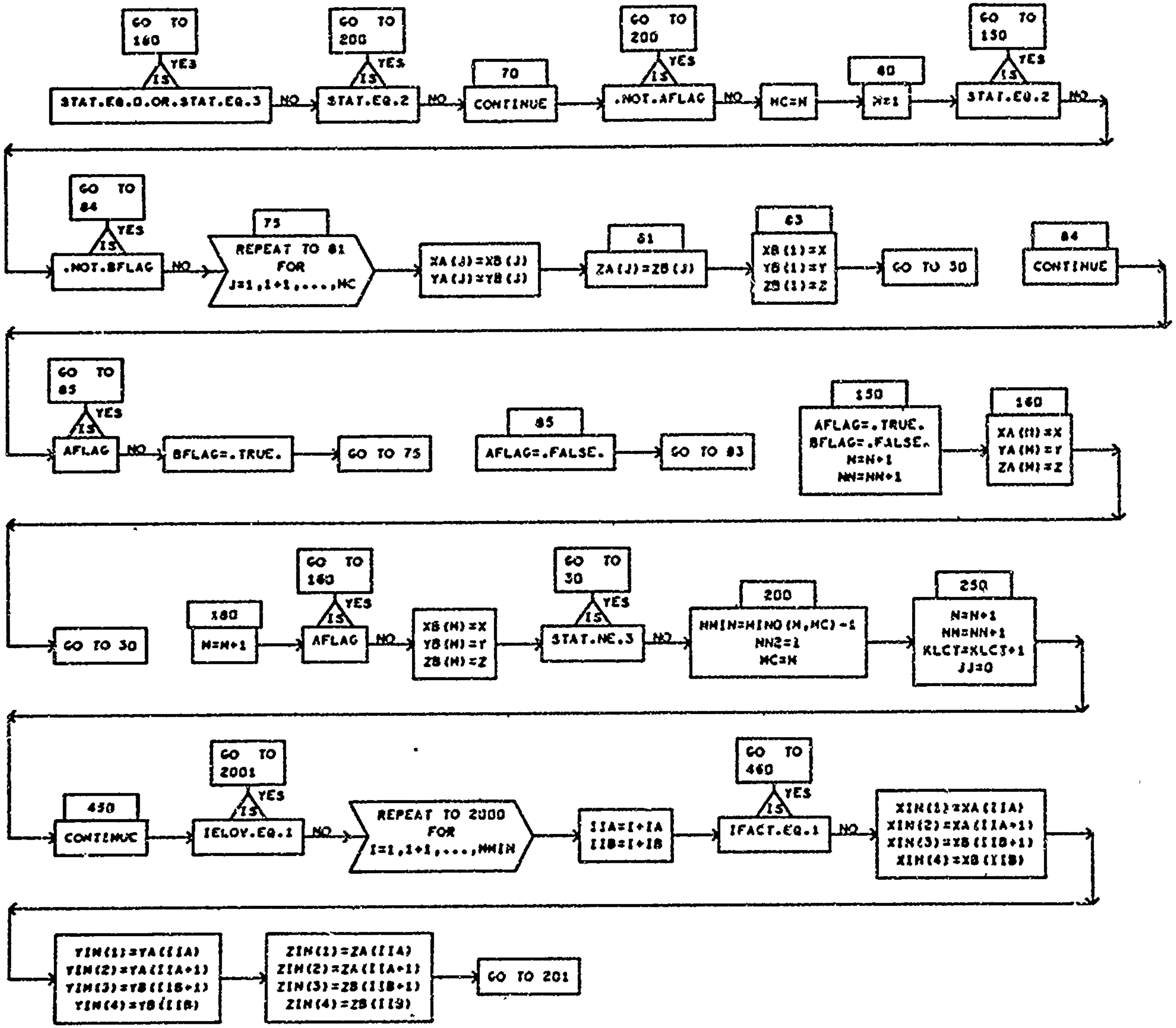
SDATA

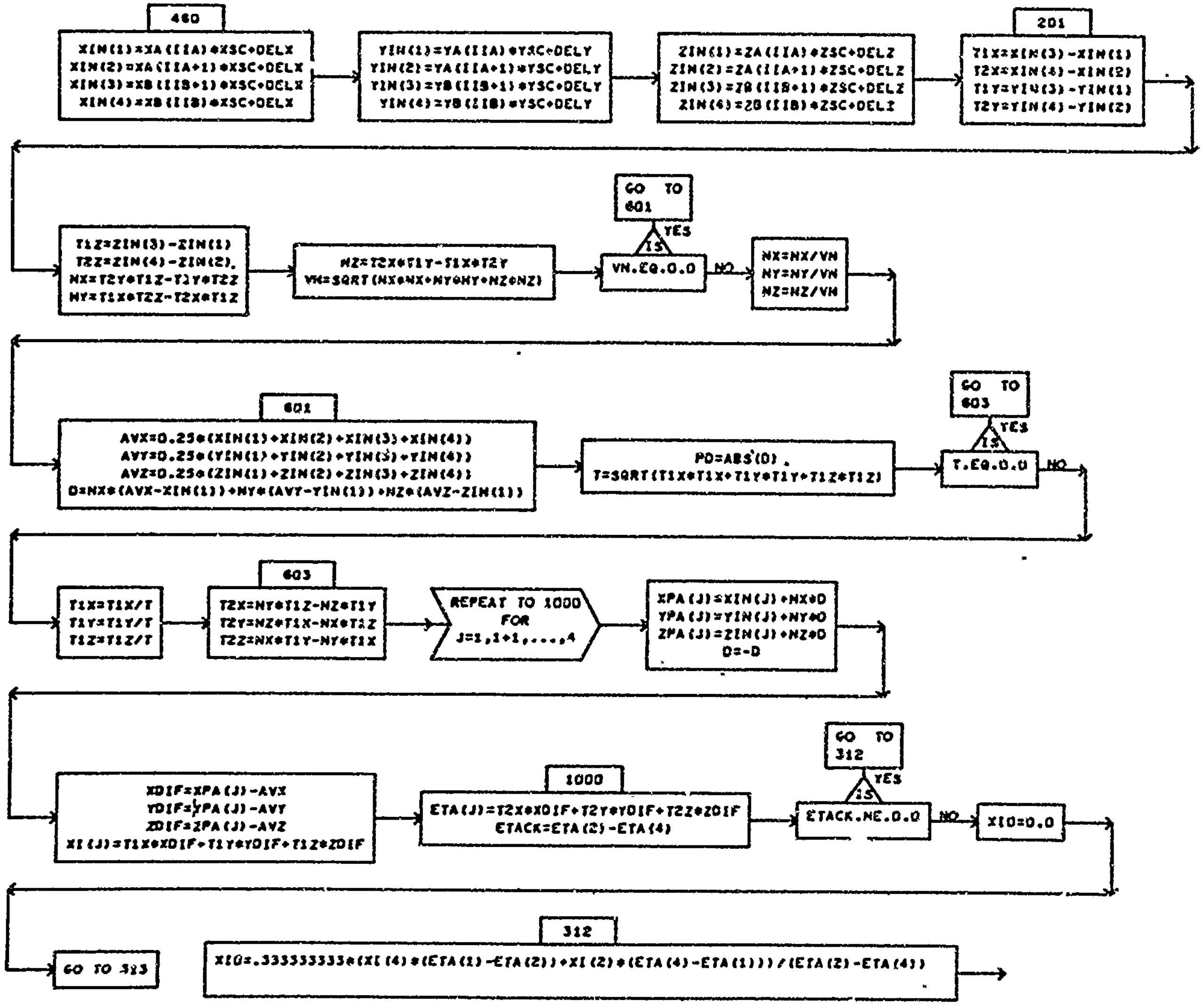


13
PERFORM SUBROUTINE ANALYS
4
CONTINUE
GO TO 6
YES
IS
ERROR.NE.0
NO
ITAPE.EQ.1.OR.ITAPE.EQ.3
YES
A5 REWIND
ITAPE.EQ.3
20
CONTINUE
ITAPE.EQ.0.OR.ITAPE.EQ.3.OR.ITAPE.EQ.4
A2 READ
ITAPE.EQ.1.OR.ITAPE.EQ.2
A3 READ
STAT=IABS(STAT)
STATT=IABS(STATT)
ITAPE.EQ.3.OR.ITAPE.EQ.4
A5 WRITE
TYPE.NE.3
SECTS=SECT
GO TO 20


GO TO 50
RFLAG
IS
YES
NO
RFLAG=.TRUE.
X=XX
Y=YY
Z=ZZ
STAY=STATT
GO TO 60
50
RFLAG=.FALSE.
ITAPE.EQ.C.OR.ITAPE.EQ.3.OR.ITAPE.EQ.4
A2 READ
READ(5,1)X,Y,Z,STAT,XX,YY,ZZ,STATT,CASE,SECT,TYPE,SEG
ITAPE.EQ.1.OR.ITAPE.EQ.2
A5 READ
READ(6,1)X,Y,Z,STAT,XX,YY,ZZ,STATT,CASE,SECT,TYPE,SEG
STAT=IABS(STAT)
STATT=IABS(STATT)
ITAPE.EQ.3.OR.ITAPE.EQ.4
A5 WRITE
WRITE(6,1)X,Y,Z,STAT,XX,YY,ZZ,STATT,CASE,SECT,TYPE,SEG
TYPE.NE.3
GO TO 300
(STAT.EQ.0.OR.STATT.EQ.0).AND.(STAT.NE.2.AND.STATT.NE.2)
SECTS=SECT
60
CONTINUE

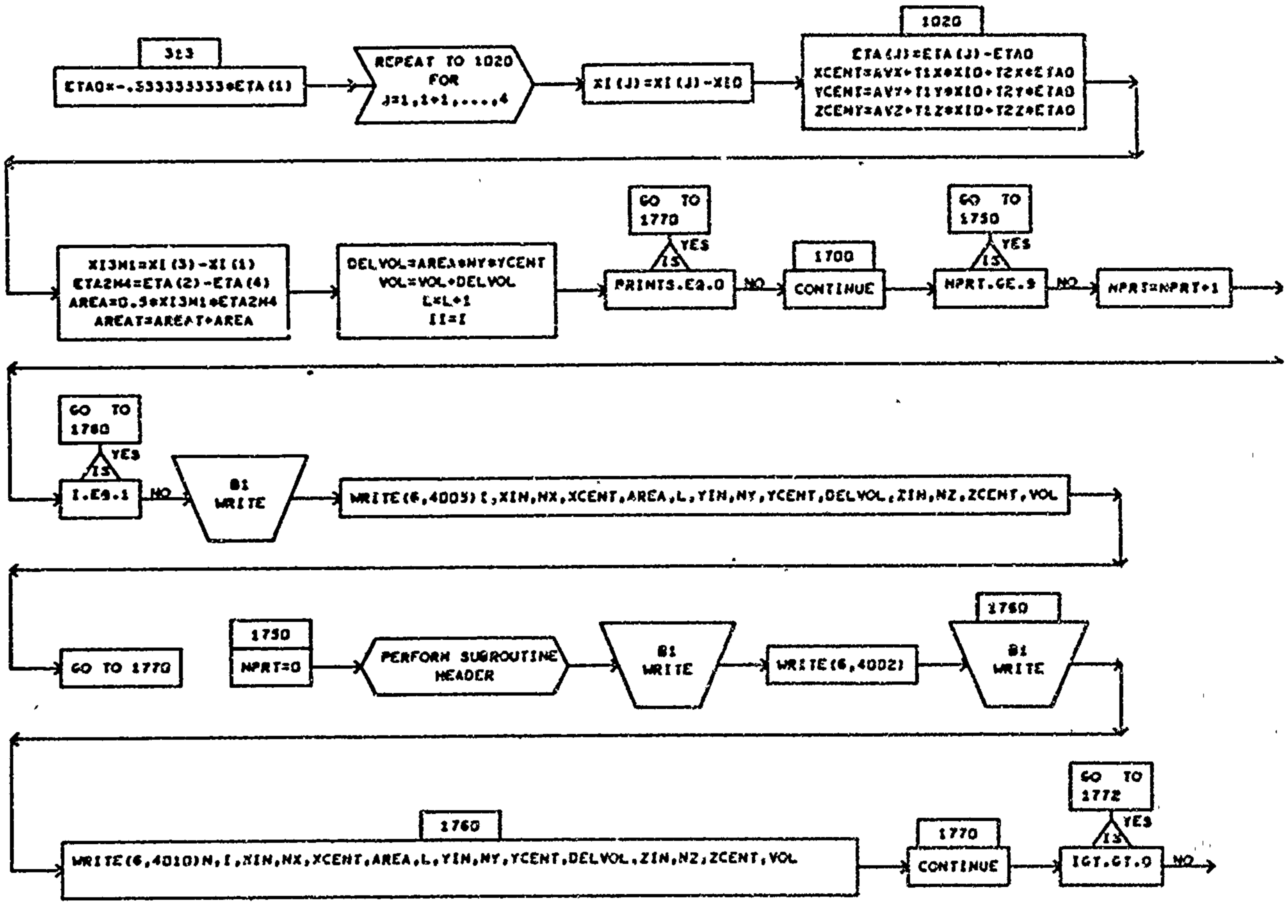
SOATA

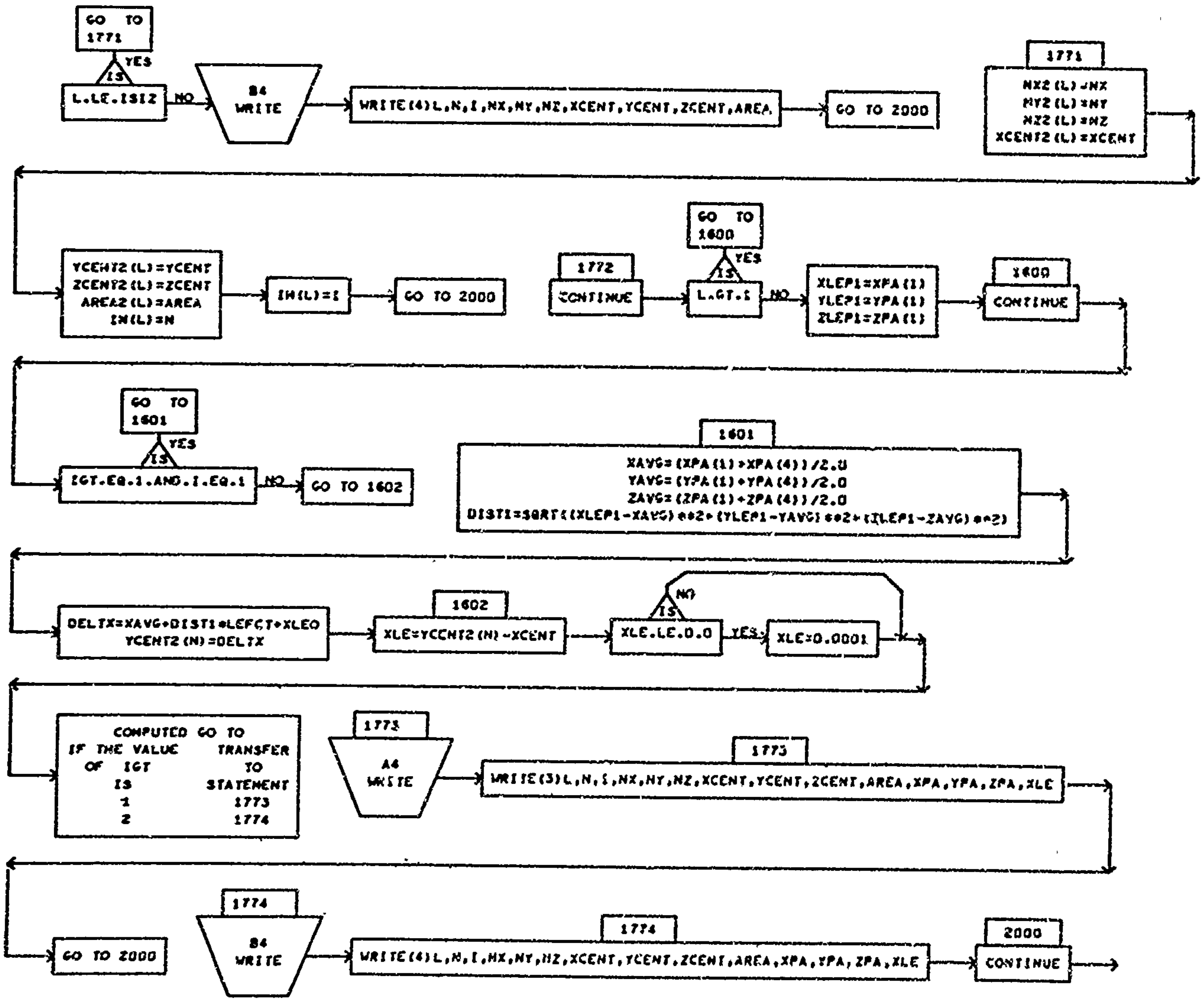

GO TO 160
YES
IS
STAT.EQ.0.OR.STAT.EQ.3
NO
GO TO 200
STAT.EQ.2
70
CONTINUE
GO TO 200
.NOT.AFLAG
MC=N
80
K=1
GO TO 150
STAT.EQ.2
GO TO 84
.NOT.BFLAG
75
REPEAT TO 81 FOR J=1,1+1,....,MC
XA(J)=XB(J)
YA(J)=YB(J)
81
ZA(J)=ZB(J)
63
XB(1)=X
YB(1)=Y
ZB(1)=Z
GO TO 30
84
CONTINUE
GO TO 85
AFLAG
BFLAG=.TRUE.
GO TO 75
85
AFLAG=.FALSE.
GO TO 83
150
AFLAG=.TRUE.
BFLAG=.FALSE.
N=N+1
NN=NN+1
160
XA(N)=X
YA(N)=Y
ZA(N)=Z
GO TO 30
180
N=N+1
GO TO 160
AFLAG
XB(N)=X
YB(N)=Y
ZB(N)=Z
GO TO 30
STAT.NE.3
200
NMIN=MINO(N,MC)-1
NN2=1
MC=N
250
N=N+1
NN=NN+1
KLCT=KLCT+1
JJ=0
450
CONTINUE
GO TO 2001
IELOV.EQ.1
REPEAT TO 2000 FOR I=1,1+1,....,NMIN
IIA=I+IA
IIB=I+IB
GO TO 460
IFACT.EQ.1
XIN(1)=XA(IIA)
XIN(2)=XA(IIA+1)
XIN(3)=XB(IIB+1)
XIN(4)=XB(IIB)
YIN(1)=YA(IIA)
YIN(2)=YA(IIA+1)
YIN(3)=YB(IIB+1)
YIN(4)=YB(IIB)
ZIN(1)=ZA(IIA)
ZIN(2)=ZA(IIA+1)
ZIN(3)=ZB(IIB+1)
ZIN(4)=ZB(IIB)
GO TO 201

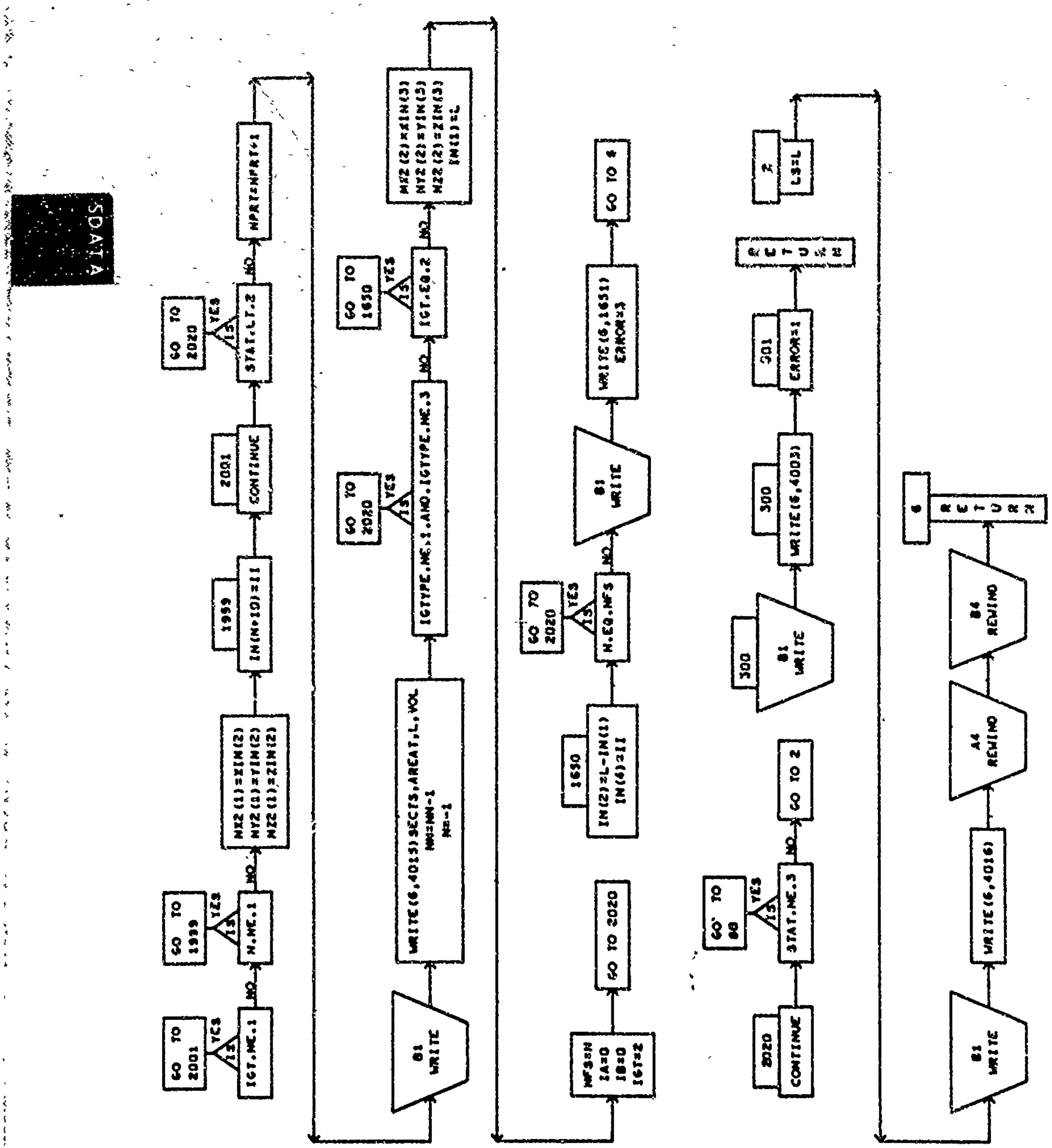

460
XIN(1)=XA(IIA)*XSC+DELX
XIN(2)=XA(IIA+1)*XSC+DELX
XIN(3)=XB(IIB+1)*XSC+DELX
XIN(4)=XB(IIB)*XSC+DELX
YIN(1)=YA(IIA)*YSC+DELY
YIN(2)=YA(IIA+1)*YSC+DELY
YIN(3)=YB(IIB+1)*YSC+DELY
YIN(4)=YB(IIB)*YSC+DELY
ZIN(1)=ZA(IIA)*ZSC+DELZ
ZIN(2)=ZA(IIA+1)*ZSC+DELZ
ZIN(3)=ZB(IIB+1)*ZSC+DELZ
ZIN(4)=ZB(IIB)*ZSC+DELZ
201
T1X=XIN(3)-XIN(1)
T2X=XIN(4)-XIN(2)
T1Y=YIN(3)-YIN(1)
T2Y=YIN(4)-YIN(2)
T1Z=ZIN(3)-ZIN(1)
T2Z=ZIN(4)-ZIN(2),
NX=T2Y*T1Z-T1Y*T2Z
NY=T1X*T2Z-T2X*T1Z
NZ=T2X*T1Y-T1X*T2Y
VN=SQRT(NX*NX+NY*NY+NZ*NZ)
GO TO 601
YES
IS
VN.EQ.0.0
NO
NX=NX/VN
NY=NY/VN
NZ=NZ/VN
601
AVX=0.25*(XIN(1)+XIN(2)+XIN(3)+XIN(4))
AVY=0.25*(YIN(1)+YIN(2)+YIN(3)+YIN(4))
AVZ=0.25*(ZIN(1)+ZIN(2)+ZIN(3)+ZIN(4))
D=NX*(AVX-XIN(1))+NY*(AVY-YIN(1))+NZ*(AVZ-ZIN(1))
PD=ABS(D),
T=SQRT(T1X*T1X+T1Y*T1Y+T1Z*T1Z)
GO TO 603
YES
IS
T.EQ.0.0
NO
T1X=T1X/T
T1Y=T1Y/T
T1Z=T1Z/T
603
T2X=NY*T1Z-NZ*T1Y
T2Y=NZ*T1X-NX*T1Z
T2Z=NX*T1Y-NY*T1X
REPEAT TO 1000
FOR
J=1,1+1,....,4
XPA(J)=XIN(J)+NX*D
YPA(J)=YIN(J)+NY*D
ZPA(J)=ZIN(J)+NZ*D
D=-D
XDIF=XPA(J)-AVX
YDIF=YPA(J)-AVY
ZDIF=ZPA(J)-AVZ
XI(J)=T1X*XDIF+T1Y*YDIF+T1Z*ZDIF
1000
ETA(J)=T2X*XDIF+T2Y*YDIF+T2Z*ZDIF
ETACK=ETA(2)-ETA(4)
GO TO 312
YES
IS
ETACK.NE.0.0
NO
XI0=0.0
GO TO 323
312
XI0=.333333333*(XI(4)*(ETA(1)-ETA(2))+XI(2)*(ETA(4)-ETA(1)))/(ETA(2)-ETA(4))

SDATA

SDATA


313
ETAO=-.333333333*ETA(1)
REPEAT TO 1020 FOR J=1,1+1,....,4
XI(J)=XI(J)-XIO
1020
ETA(J)=ETA(J)-ETAO
XCENT=AVX+T1X*XIO+T2X*ETAO
YCENT=AVY+T1Y*XIO+T2Y*ETAO
ZCENT=AVZ+T1Z*XIO+T2Z*ETAO
XI3M1=XI(3)-XI(1)
ETA2M4=ETA(2)-ETA(4)
AREA=0.5*XI3M1*ETA2M4
AREAT=AREAT+AREA
DELVOL=AREA*NY*YCENT
VOL=VOL+DELVOL
L=L+1
II=I
GO TO 1770
YES
IS
PRINTS.EQ.0
NO
1700
CONTINUE
GO TO 1750
YES
IS
NPRT.GE.9
NO
NPRT=NPRT+1
GO TO 1760
YES
IS
I.EQ.1
NO
B1
WRITE
WRITE(6,4005)I,XIN,NX,XCENT,AREA,L,YIN,NY,YCENT,DELVOL,ZIN,NZ,ZCENT,VOL
GO TO 1770
1750
NPRT=0
PERFORM SUBROUTINE HEADER
B1
WRITE
WRITE(6,4002)
1760
B1
WRITE
1760
WRITE(6,4010)N,I,XIN,NX,XCENT,AREA,L,YIN,NY,YCENT,DELVOL,ZIN,NZ,ZCENT,VOL
1770
CONTINUE
GO TO 1772
YES
IS
IGT.GT.0
NO


GO TO 1771
YES
IS
L.LE.ISIZ
NO
B4 WRITE
WRITE(4)L,N,I,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA
GO TO 2000
1771
NX2(L)=NX
NY2(L)=NY
NZ2(L)=NZ
XCENT2(L)=XCENT
YCENT2(L)=YCENT
ZCENT2(L)=ZCENT
AREA2(L)=AREA
IN(L)=N
IN(L)=I
GO TO 2000
1772
CONTINUE
GO TO 1600
YES
IS
L.GT.1
NO
XLEP1=XPA(1)
YLEP1=YPA(1)
ZLEP1=ZPA(1)
1600
CONTINUE
GO TO 1601
YES
IS
IGT.EQ.1.AND.I.EQ.1
NO
GO TO 1602
1601
XAVG=(XPA(1)+XPA(4))/2.0
YAVG=(YPA(1)+YPA(4))/2.0
ZAVG=(ZPA(1)+ZPA(4))/2.0
DISTI=SQRT((XLEP1-XAVG)**2+(YLEP1-YAVG)**2+(ZLEP1-ZAVG)**2)
DELTX=XAVG+DISTI*LEFCT+XLEO
YCENT2(N)=DELTX
1602
XLE=YCENT2(N)-XCENT
NO
IS
XLE.LE.0.0
YES
XLE=0.0001
COMPUTED GO TO
IF THE VALUE OF IGT IS 1 TRANSFER TO STATEMENT 1773
IF THE VALUE OF IGT IS 2 TRANSFER TO STATEMENT 1774
1773
A4 WRITE
1773
WRITE(3)L,N,I,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,YPA,ZPA,XLE
GO TO 2000
1774
B4 WRITE
1774
WRITE(4)L,N,I,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,YPA,ZPA,XLE
2000
CONTINUE



SDATA
GO TO 2001
IGT.NE.1
GO TO 1999
N.NE.1
NX2(1)=XIN(2)
NY2(1)=YIN(2)
NZ2(1)=ZIN(2)
1999
IN(N+10)=II
2001
CONTINUE
GO TO 2020
STAT.LT.2
NPRT=NPRT+1
B1
WRITE
WRITE(6,4015)SECTS,AREAT,L,VOL
NN=NN-1
NB=1
GO TO 2020
IGTYPE.NE.1.AND.IGTYPE.NE.3
GO TO 1650
IGT.EQ.2
NX2(2)=XIN(3)
NY2(2)=YIN(3)
NZ2(2)=ZIN(3)
IN(1)=L
NFS=N
IA=0
IB=0
IGT=2
GO TO 2020
1650
IN(2)=L-IN(1)
IN(4)=II
GO TO 2020
N.EQ.NFS
B1
WRITE
WRITE(6,1651)
ERROR=1
GO TO 4
2020
CONTINUE
GO TO 88
STAT.NE.3
GO TO 2
300
B1
WRITE
300
WRITE(6,4003)
301
ERROR=1
RETURN
2
LS=L
B1
WRITE
WRITE(6,4016)
A4
REWIND
B4
REWIND
6
RETURN

SYMBOLS USED IN SUBROUTINE SDATA

AFLAG	L	U	INPUT DATA READ CONTROL FLAG	SDATA
AREA	R	U	ELEMENT AREA	SDATA
AREAT	R	U	TOTAL AREA	SDATA
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	SDATA
AVX	R	U	AVERAGE POINT COORDINATE-X	SDATA
AVY	R	U	AVERAGE POINT COORDINATE-Y	SDATA
AVZ	R	U	AVERAGE POINT COORDINATE-Z	SDATA
BFLAG	L	U	INPUT DATA READ CONTROL FLAG	SDATA
CASE	I	C	CASE NUMBER	SDATA
D	R	U	CORNER POINT PROJECTION DISTANCE	SDATA
DELTX	R	U	DISTANCE FROM LEADING EDGE TO CENTROID	SDATA
DELVOL	R	U	ELEMENT VOLUME CONTRIBUTION	SDATA
DELX	R	U	GEOMETRY DATA X-INCREMENT	SDATA
DELY	R	U	GEOMETRY DATA Y-INCREMENT	SDATA
DELZ	R	U	GEOMETRY DATA Z-INCREMENT	SDATA
DIST1	R	U	LEADING EDGE DISTANCE VALUE	SDATA
ERROR	I	C	ERROR FLAG	SDATA
ETA	R	D	COORDINATE IN ELEMENT COORDINATE SYSTEM	SDATA
ETACK	R	U	ETA CHECK PARAMETER	SDATA
ETAO	R	U	CENTROID IN ELEMENT COORDINATE SYSTEM	SDATA
ETA2M4	R	U	CONSTANT IN AREA EQUATION	SDATA
I	I	U	ELEMENT NUMBER IN COLUMN	SDATA
IA	I	U	FLAG TO CONTROL SHIFTING OF COLUMN DATA POINTS FOR IORIEN=3	SDATA
IB	I	U	FLAG TO CONTROL SHIFTING OF COLUMN DATA POINTS FOR IORIEN=2	SDATA
IELOV	I	U	ELEMENT CHARACTERISTIC OVERRIDE FLAG	SDATA
IFACT	I	U	SCALE FACTOR FLAG	SDATA
IGEOM	I	U	GEOMETRY SOURCE FLAG	SDATA
IGT	I	U	CONTROL SURFACE FLAG	SDATA
IGTYPE	I	A	COMPONENT TYPE FLAG	SDATA
II	I	U	NUMBER OF ELEMENTS IN COLUMN	SDATA
IIA	I	U	DATA SHIFTING CONTROL PARAMETER (IORIEN=3)	SDATA
IIB	I	U	DATA SHIFTING CONTROL PARAMETER (IORIEN=2)	SDATA
IM	I	C	ELEMENT ROW NUMBER ARRAY	SDATA
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	SDATA
IORIEN	I	A	ELEMENT ORIENTATION FLAG	SDATA
ISIZ	I	A	NUMBER OF ELEMENTS TO BE STORED IN CORE	SDATA

SDATA

SYMBOLS USED IN SUBROUTINE SDATA

ITAPE	I	U	GEOMETRY TAPE CONTROL FLAG	SDATA
JJ	I	U	COUNTER	SDATA
KLCT	I	U	COUNTER	SDATA
L	I	C	NUMBER OF ELEMENTS	SDATA
LEFCT	R	U	LEADING EDGE FACTOR	SDATA
LS	I	C	NUMBER OF ELEMENTS	SDATA
M	I	U	DATA READ IN CONTROL FLAG	SDATA
MC	I	U	DATA READ IN CONTROL NUMBER	SDATA
MMIN	I	U	NUMBER OF ELEMENTS IN A COLUMN	SDATA
N	I	U	COLUMN NUMBER	SDATA
NFS	I	U	ELEMENT COUNTER	SDATA
NN	I	U	ELEMENT COUNTER	SDATA
NN2	I	U	ELEMENT COUNTER	SDATA
NPRT	I	U	LINE COUNTER	SDATA
NX	R	U	ELEMENT DIRECTION COSINE-X	SDATA
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	SDATA
NY	R	U	ELEMENT DIRECTION COSINE-Y	SDATA
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	SDATA
NZ	R	U	ELEMENT DIRECTION COSINE-Z	SDATA
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	SDATA
PAGE	I	C	PAGE NUMBER	SDATA
PD	R	U	CORNER POINT PROJECTION DISTANCE	SDATA
PRINTS	I	A	ELEMENT DATA PRINT FLAG	SDATA
RFLAG	L	U	INPUT DATA READ CONTROL FLAG	SDATA
SECT	R	U	SECTION IDENTIFICATION	SDATA
SECTS	R	U	SECTION IDENTIFICATION	SDATA
SEQ	I	U	CARD SEQUENCE NUMBER	SDATA
STAT	I	U	COORDINATE POINT STATUS FLAG	SDATA
STATT	I	U	COORDINATE POINT STATUS FLAG	SDATA
SYMFCT	I	A	SYMMETRY FLAG	SDATA
T	R	U	UNIT VECTOR	SDATA
TITLE	R	C	TITLE	SDATA
TYPE	I	U	CARD TYPE NUMBER	SDATA
T1X	R	U	X-COMPONENT OF VECTOR T1	SDATA
T1Y	R	U	Y-COMPONENT OF VECTOR T1	SDATA
T1Z	R	U	Z-COMPONENT OF VECTOR T1	SDATA

SYMBOLS USED IN SUBROUTINE SDATA

T2X	R	U	X-COMPONENT OF VECTOR T2	SDATA
T2Y	R	U	Y-COMPONENT OF VECTOR T2	SDATA
T2Z	R	U	Z-COMPONENT OF VECTOR T2	SDATA
VN	R	U	VECTOR LENGTH	SDATA
VOL	R	U	TOTAL VOLUME	SDATA
X	R	U	X-COORDINATE	SDATA
XA	R	D	X-COORDINATE	SDATA
XAVG	R	U	AVERAGE X-COORDINATE	SDATA
XB	R	D	X-COORDINATE	SDATA
XCENT	R	U	ELEMENT CENTROID COORDINATE-X	SDATA
XCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-X	SDATA
XDIF	R	U	COORDINATE DIFFERENCE-X	SDATA
XI	R	D	COORDINATE IN ELEMENT COORDINATE SYSTEM	SDATA
XIN	R	D	ELEMENT COORDINATES-X	SDATA
XIO	R	U	CENTROID IN ELEMENT COORDINATE SYSTEM	SDATA
XI3M1	R	U	CONSTANT FOR AREA EQUATION	SDATA
XLE	R	U	DISTANCE FROM LEADING EDGE TO ELEMENT CENTROID	SDATA
XLEO	R	U	LEADING EDGE X INCREMENT	SDATA
XLEP1	R	U	SAVED X-COORDINATE	SDATA
XPA	R	D	COORDINATES OF ELEMENT CORNER POINTS, X	SDATA
XSC	R	U	X SCALE FACTOR	SDATA
XX	R	U	X-COORDINATE	SDATA
Y	R	U	Y-COORDINATE	SDATA
YA	R	D	Y-COORDINATE	SDATA
YAVG	R	U	AVERAGE Y COORDINATE	SDATA
YB	R	D	Y-COORDINATE	SDATA
YCENT	R	U	ELEMENT CENTROID COORDINATE-Y	SDATA
YCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-Y	SDATA
YDIF	R	U	COORDINATE DIFFERENCE-Y	SDATA
YIN	R	D	ELEMENT COORDINATE-Y	SDATA
YLEP1	R	U	SAVED Y-COORDINATE	SDATA
YPA	R	D	COORDINATES OF ELEMENT CORNER POINTS-Y	SDATA
YSC	R	U	Y-SCALE FACTOR	SDATA
YY	R	U	Y-COORDINATE	SDATA
Z	R	U	Z-COORDINATE	SDATA
ZA	R	D	Z-COORDINATE	SDATA

SYMBOLS USED IN SUBROUTINE SDATA

ZAVG	R	U	AVERAGE Z COORDINATE	SDATA
ZB	R	D	Z-COORDINATE	SDATA
ZCENT	R	U	ELEMENT CENTROID COORDINATE-Z	SDATA
ZCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-Z	SDATA
ZDIF	R	U	COORDINATE DIFFERENCE Z	SDATA
ZIN	R	D	ELEMENT COORDINATE-Z	SDATA
ZLEP1	R	U	SAVED Z-COORDINATE	SDATA
ZPA	R	D	COORDINATES OF ELEMENT CORNER POINTS-Z	SDATA
ZSC	R	U	Z-SCALE FACTOR	SDATA
ZZ	R	U	Z-COORDINATE	SDATA

5. SUBROUTINE ANALY1 (DECK AROD)

This subprogram prepares surface-element data points for circular and elliptical cross-sections.

a. Algorithm

The Ellipse Generation Control Card (Type 4) is read as are all the Cross-Section Data Cards (Type 5). The input cross-section data are stored in the appropriate arrays. The program then takes each cross-section and calculates the surface element data as directed by the input parameters. This information is stored on Tape 11 and at the appropriate time is transmitted to the PUNCH routine, recording onto the geometry storage tape.

b. Input/Output

Ellipse Generation Control Card (Type 4), and Cross-Section Data Cards (Type 5).
No output data are produced.

c. Error

An error condition occurs when input card type numbers are wrong.

d. Subroutines Required

PUNCH

e. Argument List

None

f. Length

8552 bytes

DECK AROD   ANALY1

```
      SUBROUTINF ANALY1
C     THIS SUBROUTINE PREPARES THE REQUIRED SURFACE ELEMENTS FOR
C     CIRCULAR OR ELLIPTICAL ARC SECTIONS.  EACH CROSS-SECTION
C     IS CONSIDERED SEPARATELY. DUMMY POINTS ARE COMPUTED SO THAT EACH
C     SECTION IS FORCFD TO HAVE AN EVFN NUMBER OF POINTS AND SO THAT
C     POINTS IN A ROW ARE CORRECTLY MATCHED WITH POINTS IN AN ACJACENT
C     ROW WHEN THESE ROWS CONTAIN AN UNFQUAL NUMBER OF POINTS.
C
C  THE PARAMETER DISCON WHICH IS SPECIFIED BY THE PROGRAMMFR IS VALUED
C     DEPENDING ON HOW THE POINTS ARE TO BE MATCHED
C          DISCON= 1  ALL THETAO AND THETAL ARE THE SAME. DELTHE MUST
C                     DIVIDE THE ANGULAR INCREMENT THETAL - THETAO
C                     EVENLY.
C                = 2  ALL THETAL ARE EQUAL BUT THETAO VARIES
C                = 3  ALL THETAO ARE FQUAL BUT THFTAL VARIES
      DIMENSION TITLE(15),AX(100),THETOX(100),THETLX(100),DELTHX( 100),
     1   NN(100),SECT(1),DELZX(100),DELYX(100),AA(100),BB(100)
      COMMON CASE,TITLE,PAGE,ERROR
      INTEGER  STAT,STATT,STATD,STATC,ERROR
      INTEGER  STATA, STATB,PAGE,SEQ,TYPE,CASE,DISCON
      RADD(BBI,AAI,THP) = SQRT(BBI*BBI*COS(THP)*COS(THP) +
     1                    AAI*AAI*SIN(THP)*SIN(THP) )
      WRITE (6,603)
  603 FORMAT (1H1,/////,1H0,33HFLLIPTICAL GEOMETRY DATA IS BEING
     1 21H GENERATED **********  )
C     SFT COUNTERS
      TYPE = 3
      NREC = 0
C  READ IN TITLE CARD
    1 READ (5,600) (TITLE(L),L=1,12),DISCON,IPRINT,CASE,(SECT(L),L=1,1),
     1          ITYPE
  600 FORMAT(12A4,11X,2I1,4XI3,1A2,I2)
      IF (ITYPE .NE. 4) GC TO 700
      LINF = 100
      SEQ  = 1
```

```
DECK AROD

C  READ IN ALL DATA CARDS FOR THE SECTION                                AROD 0360
      REWIND 11                                                           AROD 0370
      I = 1                                                               AROD 0380
    4 READ (5,602) X,THETO,THETL,NN(I),A,B,DFLZ,DELY,LAST,ITYPE           AROD 0390
  602 FORMAT (F10.0,2F6.0,I3,2F10.0,2F7.0,I1,10XI2)                       AROD 0400
      IF (ITYPE .NE. 5) GC TO 700                                         AROD C410
      DELTH = (THETL-THETO)/FLOAT(NN(I))                                  AROD 0420
      THETO  = THETO /57.2957795                                          AROD 0430
      THETL  = THETL /57.2957795                                          AROD 0440
      DELTH  = DELTH /57.2957795                                          AROD 0450
      AX(I) = X                                                           AROD 0460
      THETOX(I) = THETO                                                   AROD 0470
      THETLX(I) = THETL                                                   AROD 0480
      DFLTHX(I) = DELTH                                                   AROD 0490
      AA(I) = A                                                           AROD 0500
      BB(I) = B                                                           AROD 0510
      DELYX(I) = DELY                                                     AROD 0520
      DELZX(I) = DELZ                                                     AROD 0530
      IF (LAST .EQ. 0)  GO TO 2                                           AROD C540
      N = I                                                               AROD 0550
      GO TO 3                                                             AROD 0560
    2 I = I + 1                                                           AROD 0570
      GO TO 4                                                             AROD 0580
    3 I = 1                                                               AROD 0590
      M = 0                                                               AROD 0600
    8 IF (I .GT. N)  GO TO 5                                              AROD 0610
C                                                                         AROD 0620
      IF (NN(I)-M)  6,6,7                                                 AROD 0630
    7 M =NN(I)                                                            AROD 0640
    6 I = I + 1                                                           AROD 0650
      GO TO 8                                                             AROD 0660
C                                                                         AROD 0670
    5 GO TO (10C,2C0,3C0), DISCON                                         AROD 0680
C                                                                         AROD C690
  100 M = M + 1                                                           AROD 0700
C                                                                         AROD 0710
```

DECK AROD

```
      DO 1C1 I=1,N
C
      DO 1C2 J=1,M
C
      XA = AX(I)
      THETA = THETOX(I) + (FLOAT(J-1)) * DELTHX(I)
      THETAP = ABS(THETA - 1.57079633)
      RAD = RADD(BB(I),AA(I),THETAP)
      IF (RAD .NE. 0.0)  RAD = AA(I)*BB(I) / RAD
      YA = RAD * SIN(THETA)
      ZA =-RAD * COS(THETA)
      YA = YA + DELYX(I)
      ZA = ZA + DELZX(I)
C
      IF (J .EQ. 1) GO TO 1C3
      STATA = 0
      GO TO 105
C
  103 IF (I .EQ. 1) GO TO 104
      STATA = 1
      GO TO 105
C
  104 STATA = 2
  105 WRITE (11)XA,YA,ZA,STATA,STATA
  102 CONTINUE
  101 CONTINUE
C
      GO TO 10
C
C
  200 DO 201 I=1,N
C
      LIM = M+1-NN(I)
C
      DO 202 J=1,LIM
      XA = AX(I)
```

DECK AROD

```
      THETA = THETOX(I)                                               AROD 1080
      THETAP = ABS(THETA - 1.57079633)                                AROD 1090
      RAD = RADD(BB(I),AA(I),THETAP)                                  AROD 1100
      IF (RAD .NE. 0.0)  RAD = AA(I)*BB(I) / RAD                      AROD 1110
      YA = RAD * SIN(THETA)                                           AROD 1120
      ZA =-RAD * COS(THETA)                                           AROD 1130
      YA = YA + DELYX(I)                                              AROD 1140
      ZA = ZA + DELZX(I)                                              AROD 1150
      IF ( J .EQ. 1)  GO TO 203                                       AROD 1160
      STATA = 0                                                       AROD 1170
      GO TO 205                                                       AROD 1180
C                                                                     AROD 1190
  203 IF (I .EQ. 1)  GO TO 204                                        AROD 1200
      STATA = 1                                                       AROD 1210
      GO TO 205                                                       AROD 1220
C                                                                     AROD 1230
  204 STATA = 2                                                       AROD 1240
  205 WRITE (11)XA,YA,ZA,STATA,STATA                                  AROD 1250
C                                                                     AROD 1260
  202 CONTINUE                                                        AROD 1270
      K=0                                                             AROD 1280
      LIM = LIM +1                                                    AROD 1290
      NM =M * 2                                                       AROD 1300
      DO 206  J = LIM,NM                                              AROD 1310
      XA = AX(I)                                                      AROD 1320
      THETA = THETLX(I)-(FLOAT(NN(I)-K))*DELTHX(I)                    AROD 1330
      THETAP = ABS(THETA - 1.57079633)                                AROD 1340
      RAD = RADD(BB(I),AA(I),THETAP)                                  AROD 1350
      IF (RAD .NE. 0.0)  RAD = AA(I)*BB(I) / RAD                      AROD 1360
      YA = RAD * SIN(THETA)                                           AROD 1370
      ZA =-RAD * COS(THETA)                                           AROD 1380
      YA = YA + DELYX(I)                                              AROD 1390
      ZA = ZA + DELZX(I)                                              AROD 1400
C                                                                     AROD 1410
      IF (J.EQ. 1)  GO TO 207                                         AROD 1420
      STATA = 0                                                       AROD 1430
```

DECK AROD

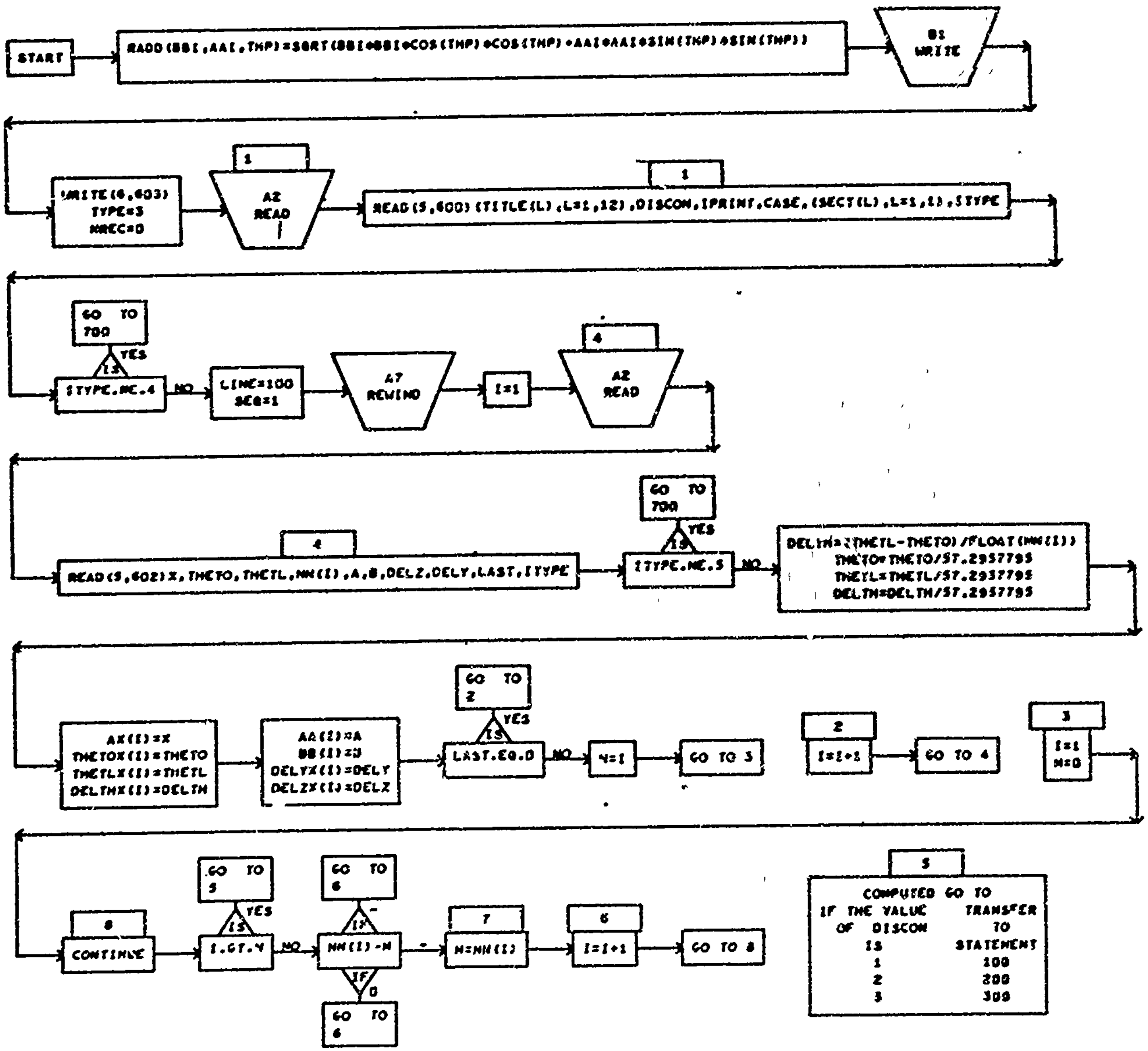
```
      GO TO 2C8                                                         AROD 1440
C                                                                       AROD 1450
  207 STATA = 1                                                         AROD 1460
C                                                                       AROD 1470
 2C8  WRITE (11)XA,YA,ZA,STATA,STATA                                    AROD 1480
      K = K + 1                                                         AROD 1490
  2C6 CONTINUE                                                          AROD 1500
C                                                                       AROD 1510
  201 CONTINUE                                                          AROD 1520
      M = M + 2                                                         AROD 1530
      GO TO 10                                                          AROD 1540
C                                                                       AROD 1550
  300 M = M +2                                                          AROD 1560
      DO 301 I = 1,N                                                    AROD 1570
      NM =NN(I) +1                                                      AROD 1580
C                                                                       AROD 1590
      DO 302 J = 1,NM                                                   AROD 1600
      XA = AX(I)                                                        AROD 1610
      THETA = THETOX(I) + (FLCAT(J-1))*DELTHX(I)                        AROD 1620
      THETAP = ABS(THETA - 1.57079633)                                  AROD 1630
      RAD = RADD(BB(I),AA(I),THETAP)                                    AROD 1640
      IF (RAD .NE. 0.0)  RAD = AA(I)*BB(I) / RAD                        AROD 1650
      YA = RAD * SIN(THETA)                                             AROD 1660
      ZA =-RAD * COS(THETA)                                             AROD 1670
      YA = YA + DELYX(I)                                                AROD 1680
      ZA = ZA + DELZX(I)                                                AROD 1690
      IF (J.EQ.1) GO TO 303                                             AROD 1700
      STATA = 0                                                         AROD 1710
      GO TO 305                                                         AROD 1720
C                                                                       AROD 1730
  3C3 IF (I.EQ.1) GO TO 304                                             AROD 1740
      STATA = 1                                                         AROD 1750
      GO TO 305                                                         AROD 1760
C                                                                       AROD 1770
  304 STATA = 2                                                         AROD 1780
  305 WRITE (11)XA,YA,ZA,STATA,STATA                                    AROD 1790
```

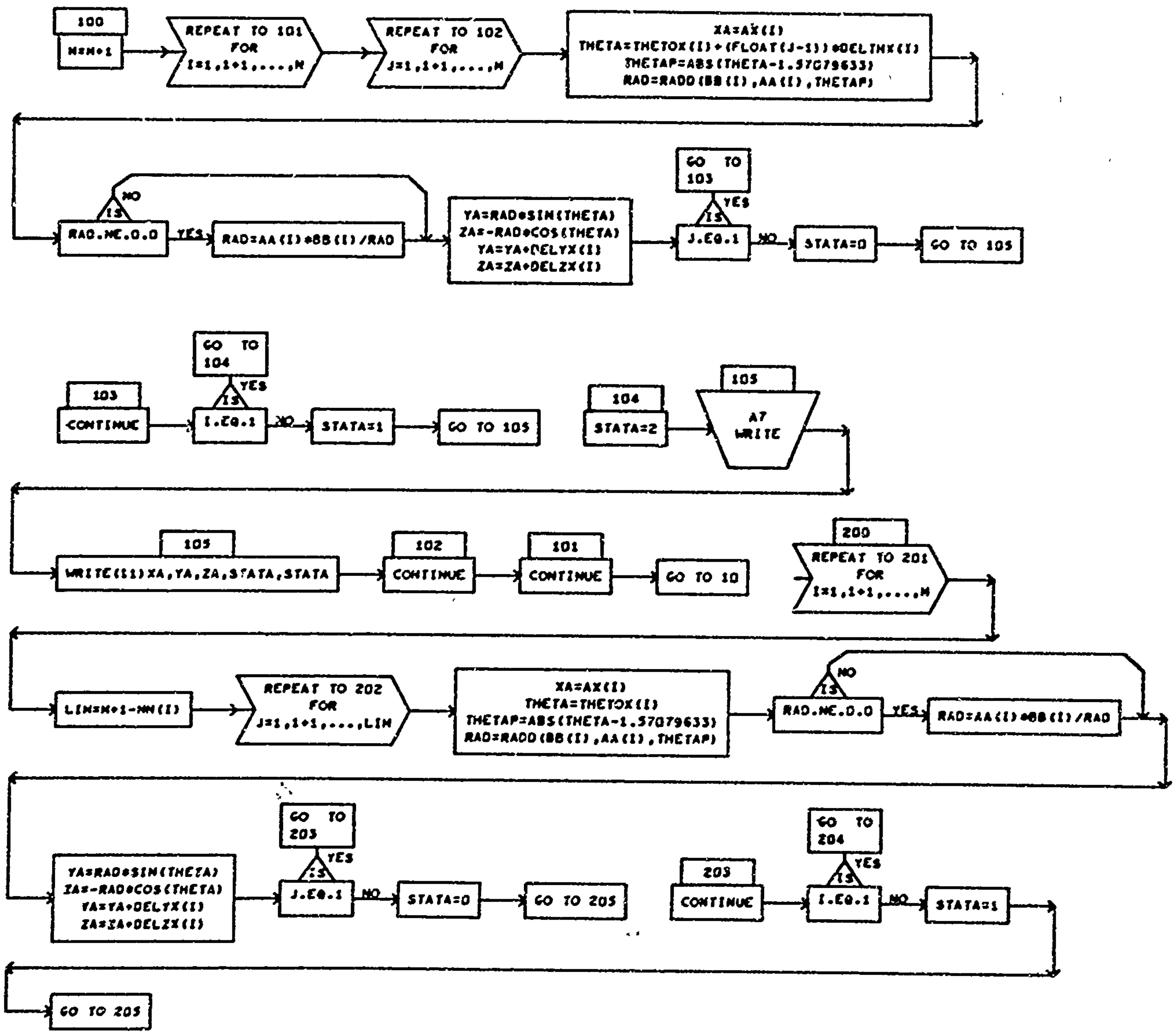
```
DECK AROD
C
  302 CONTINUE
      NM =NM+1
C
      DO 306  J = NM,M
      XA = AX(I)
      THETA = THETLX(I)
      THETAP = ABS(THETA - 1.57079633)
      RAD = RADD(BB(I),AA(I),THETAP)
      IF (RAD .NE. 0.0)  RAD = AA(I)*BB(I) / RAD
      YA = RAD * SIN(THETA)
      ZA =-RAD * COS(THETA)
      YA = YA + DELYX(I)
      ZA = ZA + DELZX(I)
      IF (J.EQ.1) GO TO 307
      STATA = 0
      GO TO 308
C
  307 STATA = 1
  308 WRITE (11)XA,YA,ZA,STATA,STATA
C
  306 CONTINUE
C
  301 CONTINUE
C
C
   10 STATA = 3
      IF (LAST.EQ.0 .OR. LAST.EQ.2) STATA = 4
C
      BACKSPACE 11
      READ (11)XA, YA, ZA, STAT,STAT
      BACKSPACE 11
      WRITE (11)XA, YA, ZA, STATA ,STATA
C
      REWIND 11
      K = 1
```

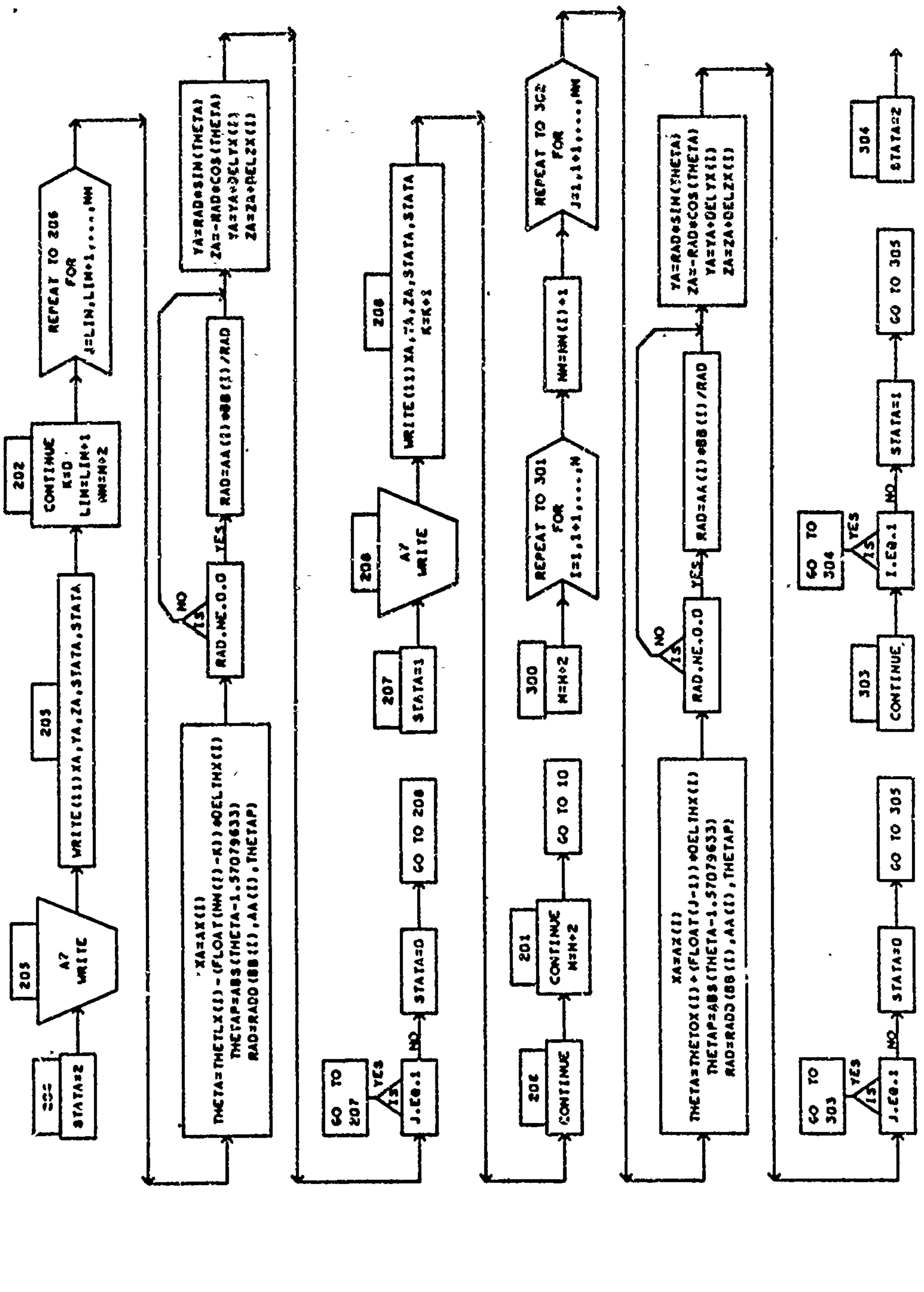
```
DECK AROD
C                                                                         AROD  2160
   15 READ (11) X,Y,Z,STAT,STAT                                           AROD  2170
C                                                                         AROD  2180
      IF ( STAT .GT. 2)      GO TO 13                                     AROD  2190
      READ (11) XX,YY,ZZ,STATT,STATT                                      AROD  2200
C                                                                         AROD  2210
      IF ( STATT .GT. 2)     GO TO 14                                     AROD  2220
C                                                                         AROD  2230
  17  CALL PUNCH (X,Y,Z,STAT,XX,YY,ZZ,STATT,SECT,TYPE,LINE,SEQ,           AROD  2240
     1  LAST,IPRINT,NREC)                                                 AROD  2250
C                                                                         AROD  2260
      GO TO (15,1,18), K                                                  AROD  2270
C                                                                         AROD  2280
C                                                                         AROD  2290
   14  IF (STATT.EQ. 3) GO TO 16                                          AROD  2300
      STATT = 0                                                           AROD  2310
      K = 2                                                               AROD  2320
      GO TO 17                                                            AROD  2330
C                                                                         AROD  2340
   16 K = 3                                                               AROD  2350
      GO TO 17                                                            AROD  2360
C                                                                         AROD  2370
C                                                                         AROD  2380
   13 XB = X                                                              AROD  2390
      YB = Y                                                              AROD  2400
      ZB = Z                                                              AROD  2410
      STATB = STAT                                                        AROD  2420
      GO TO 21                                                            AROD  2430
  20  BACKSPACE 11                                                        AROD  2440
   21 BACKSPACE 11                                                        AROD  2450
      READ (11)XA,YA,ZA,STATA ,STATA                                      AROD  2460
      IF (STATA.EQ.1 .OR. STATA.EQ.2) GO TO 22                            AROD  2470
      GO TO 20                                                            AROD  2480
  22  STATC = 0                                                           AROD  2490
      CALL PUNCH (XB,YB,ZB,STATC,XA,YA,ZA,STATA,SECT,TYPE,LINE,SEQ,       AROD  2500
     1      LAST,IPRINT,NREC)                                             AROD  2510
```

```
DECK AROD

      STATD = 0                                                         AROD  2520
      IF (STAT .EQ. 3)  STATD = 3                                       AROD  2530
      READ (11)XC, YC, ZC, STATT ,STATT                                 AROD  2540
      READ (11) XD, YD, ZD, STATT ,STATT                                AROD  2550
      CALL PUNCH (XC,YC,ZC,STATC,XD,YD,ZD,STATD,SECT,TYPE,LINE,SEQ,     AROD  2560
     1     LAST,IPRINT,NREC)                                            AROD  2570
                                                                        AROD  2580
      IF (STAT .NE. 3)  GO TO 1                                         AROD  2590
C                                                                       AROD  2600
C                                                                       AROD  2610
   18 CONTINUE                                                          AROD  2620
      WRITE (8,500)                                                     AROD  2630
  500 FORMAT (12H**BLANK CARD,68X)                                      AROD  2640
      END FILE 8                                                        AROD  2650
      BACKSPACE 8                                                       AROD  2660
      BACKSPACE 8                                                       AROD  2670
      IF (LAST .NE. 3) GO TO 1000                                       AROD  2680
      DO 23 I=1,NREC                                                    AROD  2690
   23 BACKSPACE 8                                                       AROD  2700
 1000 RETURN                                                            AROD  2710
  700 ERROR = 1                                                         AROD  2720
      RETURN                                                            AROD  2730
      END                                                               AROD  2740
```

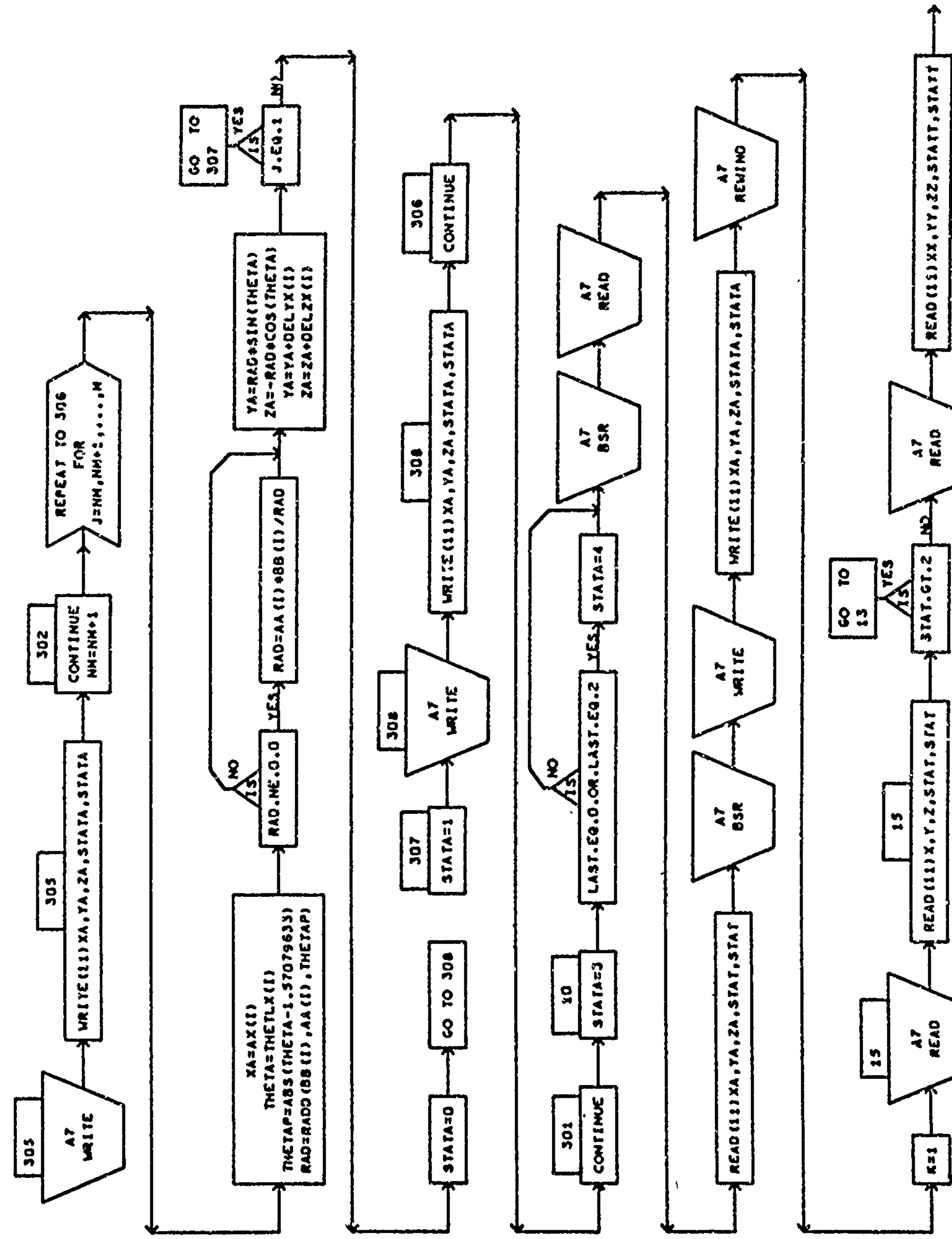
SUBROUTINE ANALYS

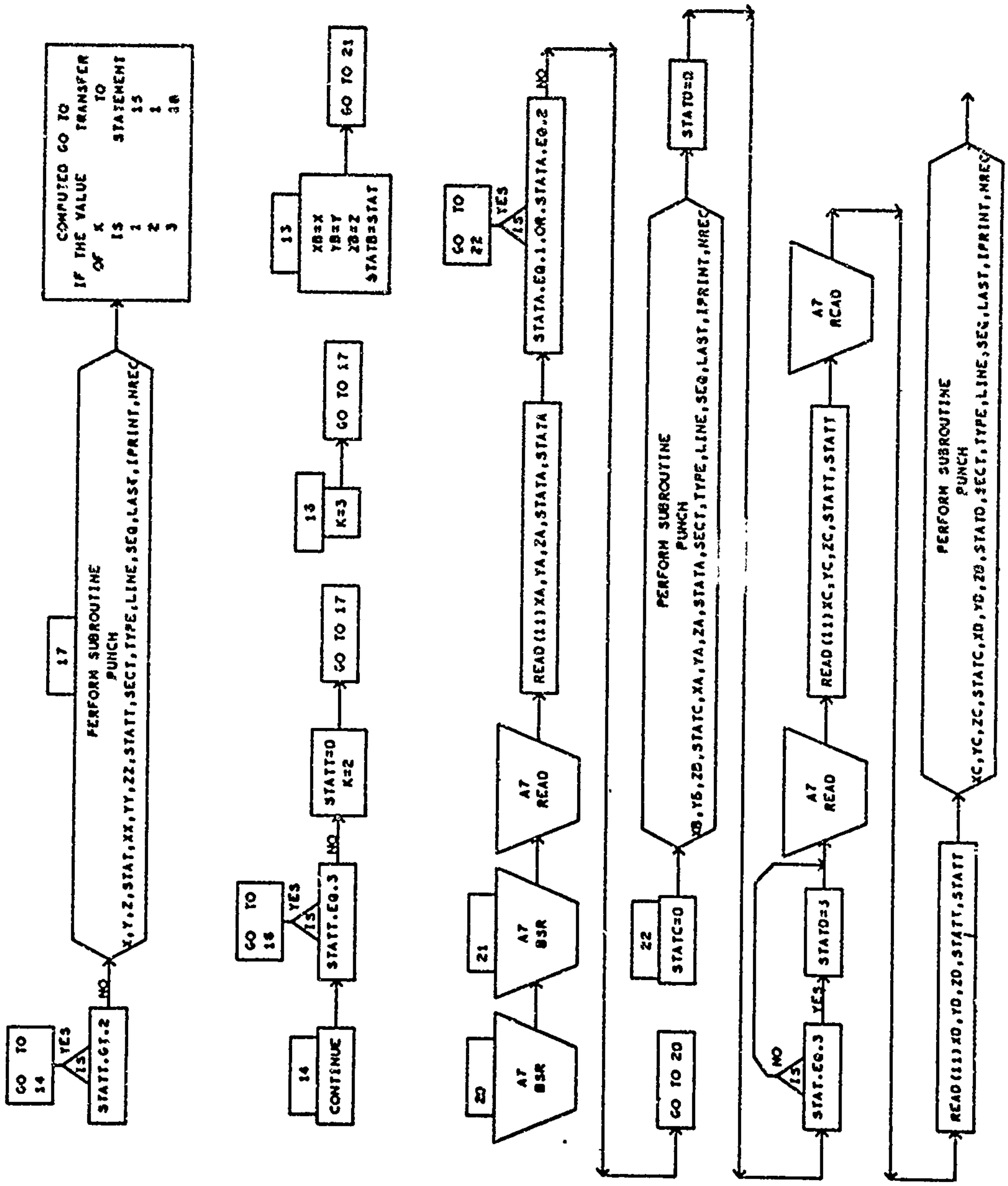


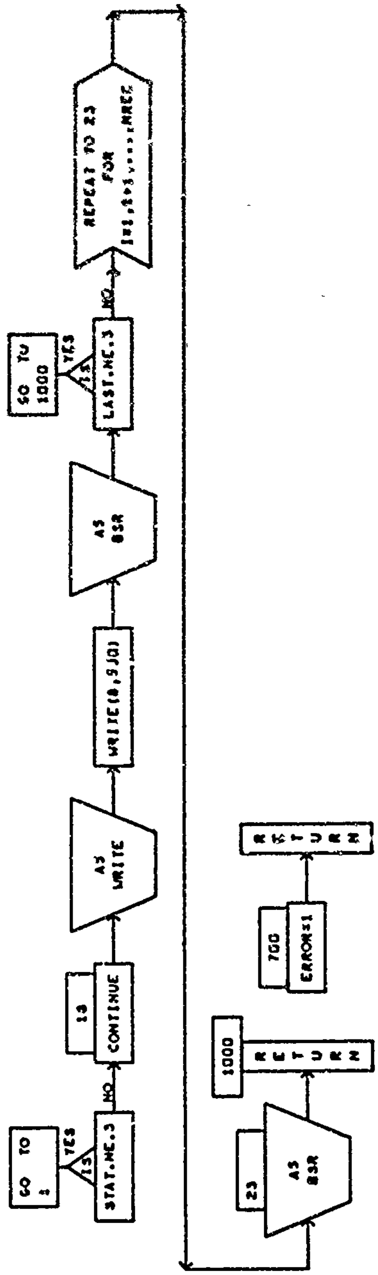

100
N=N+1
REPEAT TO 101 FOR I=1,1+1,....,N
REPEAT TO 102 FOR J=1,1+1,....,M
XA=AX(I)
THETA=THETOX(I)+(FLOAT(J-1))*DELTHX(I)
THETAP=ABS(THETA-1.57079633)
RAD=RADD(BB(I),AA(I),THETAP)
RAD.NE.0.0
IS
NO
YES
RAD=AA(I)*BB(I)/RAD
YA=RAD*SIN(THETA)
ZA=-RAD*COS(THETA)
YA=YA+DELYX(I)
ZA=ZA+DELZX(I)
J.EQ.1
GO TO 103
STATA=0
GO TO 105
103
CONTINUE
I.EQ.1
GO TO 104
STATA=1
GO TO 105
104
STATA=2
105
A7
WRITE
105
WRITE(11)XA,YA,ZA,STATA,STATA
102
CONTINUE
101
CONTINUE
GO TO 10
200
REPEAT TO 201 FOR I=1,1+1,....,N
LIN=N+1-NN(I)
REPEAT TO 202 FOR J=1,1+1,....,LIN
XA=AX(I)
THETA=THETOX(I)
THETAP=ABS(THETA-1.57079633)
RAD=RADD(BB(I),AA(I),THETAP)
RAD.NE.0.0
RAD=AA(I)*BB(I)/RAD
YA=RAD*SIN(THETA)
ZA=-RAD*COS(THETA)
YA=YA+DELYX(I)
ZA=ZA+DELZX(I)
J.EQ.1
GO TO 203
STATA=0
GO TO 205
203
CONTINUE
I.EQ.1
GO TO 204
STATA=1
GO TO 205



STATA=2
A7 WRITE
205
WRITE(11)XA,YA,ZA,STATA,STATA,STATA
202
CONTINUE
K=0
LIN=LIN+1
NN=N+2
REPEAT TO 206 FOR J=LIN,LIN+1,...,NN
XA=AX(I)
THETA=THETLX(I)-(FLOAT(NN(I)-K))*DELTHX(I)
THETAP=ABS(THETA-1.57079633)
RAD=RADD(BB(I),AA(I),THETAP)
RAD.NE.0.0
IS
NO
YES
RAD=AA(I)*BB(I)/RAD
YA=RAD*SIN(THETA)
ZA=-RAD*COS(THETA)
YA=YA+DELYX(I)
ZA=ZA+DELZX(I)
GO TO 207
J.EQ.1
STATA=0
GO TO 208
207
STATA=1
208
A7 WRITE
206
WRITE(11)XA,YA,ZA,STATA,STATA,STATA
K=K+1
CONTINUE
201
CONTINUE
N=N+2
GO TO 10
300
N=N+2
REPEAT TO 301 FOR I=1,I+1,...,N
NN=NN(I)+1
REPEAT TO 302 FOR J=1,J+1,...,NN
XA=AX(I)
THETA=THETOX(I)+(FLOAT(J-1))*DELTHX(I)
THETAP=ABS(THETA-1.57079633)
RAD=RADD(BB(I),AA(I),THETAP)
GO TO 303
303
CONTINUE
I.EQ.1
GO TO 304
GO TO 305
STATA=1
304
STATA=2


305
A7 WRITE
305
WRITE(11)XA,YA,ZA,STATA,STATA
302
CONTINUE NM=NM+1
REPEAT TO 306 FOR J=NM,NM+1,...,M
XA=AX(I)
THETA=THETLX(I)
THETAP=ABS(THETA-1.5707963)
RAD=RADD(BB(I),AA(I),THETAP)
IS
RAD.NE.0.0
NO
YES
RAD=AA(I)*BB(I)/RAD
YA=RAD*SIN(THETA)
ZA=-RAD*COS(THETA)
YA=YA+DELYX(I)
ZA=ZA+DELZX(I)
IS
J.EQ.1
YES
GO TO 307
NO
STATA=0
GO TO 308
307
STATA=1
308
A7 WRITE
308
WRITE(11)XA,YA,ZA,STATA,STATA
306
CONTINUE
301
CONTINUE
10
STATA=3
IS
LAST.EQ.0.OR.LAST.EQ.2
NO
YES
STATA=4
A7 BSR
A7 READ
READ(11)XA,YA,ZA,STAT,STAT
A7 BSR
A7 WRITE
WRITE(11)XA,YA,ZA,STATA,STATA
A7 REWIND
K=1
15
A7 READ
15
READ(11)X,Y,Z,STAT,STAT
IS
STAT.GT.2
YES
GO TO 13
NO
A7 READ
READ(11)XX,YY,ZZ,STATT,STATT




GO TO 1
IS
YES
STAT.NE.3
NO
13
CONTINUE
AS WRITE
AS BSR
GO TO 1000
YES
IS
LAST.NE.3
NO
REPEAT TO 23 FOR
23
AS BSR
1000
RETURN
700
ERROR=1
RETURN

SYMBOLS USED IN SUBROUTINE ANALY1

A	R	U	ELLIPSE RADIUS IN Y-DIRECTION	ANALY1
AA	R	D	Y-DIRECTION ELLIPSE RADIUS ARRAY	ANALY1
AX	R	D	X-STATION ARRAY	ANALY1
B	R	U	ELLIPSE RADIUS IN Z-DIRECTION	ANALY1
BB	R	D	Z-DIRECTION ELLIPSE RADIUS ARRAY	ANALY1
CASE	I	C	CASE NUMBER	ANALY1
DELTH	R	U	THETA ANGULAR INCREMENT	ANALY1
DELTHX	R	D	THETA ANGULAR INCREMENT ARRAY	ANALY1
DELY	R	U	Y-SHIFT INCREMENT FOR ELLIPSE DATA	ANALY1
DELYX	R	D	Y-SHIFT INCREMENT ARRAY	ANALY1
DELZ	R	U	Z-SHIFT INCREMENT FOR ELLIPSE DATA	ANALY1
DELZX	R	D	Z-SHIFT INCREMENT ARRAY	ANALY1
DISCON	I	U	ANGULAR MODE CONTROL FLAG	ANALY1
ERROR	I	C	ERROR FLAG	ANALY1
I	I	U	SECTION INDEX NUMBER	ANALY1
IPRINT	I	U	PRINT FLAG	ANALY1
ITYPE	I	U	CARD TYPE	ANALY1
J	I	U	ANGULAR INCREMENT COUNTER	ANALY1
K	I	U	CARD WRITE CYCLE FLAG	ANALY1
LAST	I	U	LAST CROSS-SECTION FLAG	ANALY1
LIM	I	U	ANGULAR LIMIT FLAG	ANALY1
LINE	I	U	LINE COUNTER	ANALY1
M	I	U	COUNTER	ANALY1
N	I	U	COUNTER	ANALY1
NM	I	U	COUNTER	ANALY1
NN	I	D	COUNTER	ANALY1
NREC	I	U	TAPE 8 RECORD COUNTER	ANALY1
PAGE	I	C	PAGE NUMBER	ANALY1
RAD	R	U	ELLIPSE RADIUS	ANALY1
SECT	R	D	SECTION IDENTIFICATION	ANALY1
SEQ	I	U	CARD SEQUENCE NUMBER	ANALY1
STAT	I	U	POINT STATUS FLAG	ANALY1
STATA	I	U	POINT STATUS FLAG	ANALY1
STATB	I	U	POINT STATUS FLAG	ANALY1
STATC	I	U	POINT STATUS FLAG	ANALY1
STATD	I	U	POINT STATUS FLAG	ANALY1

```
SYMBOLS USED IN SUBROUTINE ANALY1

STATT    I  U  POINT STATUS FLAG                                ANALY1
THETA    R  U  ANGULAR POSITION                                 ANALY1
THETAP   R  U  ANGULAR POSITION                                 ANALY1
THETL    R  U  LAST THETA ANGLE                                 ANALY1
THETLX   R  D  LAST THETA ANGLE ARRAY                           ANALY1
THETO    R  U  INITIAL THETA                                    ANALY1
THETOX   R  D  INITIAL THETA ANGLE ARRAY                        ANALY1
TITLE    R  C  TITLE                                            ANALY1
TYPE     I  U  CARD TYPE FOR GEOMETRY DATA =3                   ANALY1
X        R  U  X-COORDINATE                                     ANALY1
XA       R  U  X-COORDINATE                                     ANALY1
XB       R  U  X-COORDINATE                                     ANALY1
XC       R  U  X-COORDINATE                                     ANALY1
XD       R  U  X-COORDINATE                                     ANALY1
XX       R  U  X-COORDINATE                                     ANALY1
Y        R  U  Y-COORDINATE                                     ANALY1
YA       R  U  Y-COORDINATE                                     ANALY1
YB       R  U  Y-COORDINATE                                     ANALY1
YC       R  U  Y-COORDINATE                                     ANALY1
YD       R  U  Y-COORDINATE                                     ANALY1
YY       R  U  Y-COORDINATE                                     ANALY1
Z        R  U  Z-COORDINATE                                     ANALY1
ZA       R  U  Z-COORDINATE                                     ANALY1
ZB       R  U  Z-COORDINATE                                     ANALY1
ZC       R  U  Z-COORDINATE                                     ANALY1
ZD       R  U  Z-COORDINATE                                     ANALY1
ZZ       R  U  Z-COORDINATE                                     ANALY1
```

6. SUBROUTINE ANALY2 (DECK AROE)

This routine prepares surface element data from parametric cubic boundary patch data.

a. Algorithm

The routine first reads in the Parametric Cubic Title Card (Type 6). The boundary curve data for a patch is then read in (Type 7 cards). A maximum of 20 points per boundary are permitted. The boundary curve coordinate arrays are set up and the arc length along each boundary calculated. The tangent vectors at the corner points and the related end-point derivatives are calculated. The constants for the boundary curve equations are then determined. The routine then starts a cycle to generate element data points on the surface of the patch. This requires the determination of the blending functions, the related points on the boundary curves, and finally, the solution of the surface equation. The element data are transmitted to the PUNCH routine where they are recorded onto Tape 8.

b. Input/Output

Parametric Cubic Title Card (Type 6), Parametric Cubic Boundary Data Cards (Type 7).
No output data are produced.

c. Error

An error condition occurs when input card type numbers are wrong.

d. Subroutines Required

PUNCH

e. Argument List

None

f. Length

6688 bytes

DECK AROE

```
      SUBROUTINE ANALY2
C THIS SUBPROGRAM CALCULATES THE QUADRILATERAL DATA FOR A SURFACE GIVFN
C BY THE COONS MIT SURFACE FIT TECHNIQUE.
C
      DIMENSION XA(20),XB(20),YA(20),YB(20),ZA(20),ZB(20),XB1(4,20),YB1(
     14,20),ZB1(4,20),NPTS(4),D(4,9),TITLE(15),SECT(1)
      COMMON CASE,TITLE,PAGE,ERROR
      REAL L21,L31,L32,L1,M1,N1,L2,M2,N2,LN,MN,NN
      INTEGER STAT,STATT,TYPE,SEQ,CASE,PAGF,ERROR
      WRITE (6,200)
  200 FORMAT (1H1,///////,1H0,33HPARAMETRIC CUBIC GEOMETRY DATA IS
     1     24H BEING GENERATED *******  )
C
C
      TYPE=3
      SEQ=1
      LINE=100
      NREC=0
C
C *******RFAD IN BOUNDARY CURVE DATA
C     SET UP STARTING CONSTANTS
C
  100 CONTINUE
      N=-1
      L=C
    1 II=1
      ITRUE=0
      IFALSE=1
      READ (5,96)(TITLE(K),K=1,12),NCU,NOW,LAST,ISOVR,IPRINT,CASE,
     1     (SECT(K),K=1,1),ITYPE
   96 FORMAT(12A4,1X,I3,1X,I3,3X,3I1,3XI3,1A2,I2)
      IF (ITYPE .NE. 6) GC TO 51
C
C     READ IN BOUNDARY CURVE DATA
C
```

DECK AROE

```
   29 CONTINUE                                                      AROE 0360
      READ (5,61)X,Y,Z,ISTAT,XX,YY,ZZ,ISTATT,ITYPE                  AROE 0370
   61 FORMAT(3F10.4,I1,3F10.4,I1,8XI2)                              AROE 0380
      IF (ITYPE .NE. 7) GO TO 51                                    AROE 0390
      IRFLAG=IFALSE                                                 AROE 0400
      GO TO 80                                                      AROE 0410
   30 IF(IRFLAG)10,50,10                                            AROE 0420
   10 IRFLAG=ITRUE                                                  AROE 0430
      X=XX                                                          AROE 0440
      Y=YY                                                          AROE 0450
      Z=ZZ                                                          AROE 0460
      ISTAT=ISTATT                                                  AROE 0470
      GO TO 60                                                      AROE 0480
   50 IRFLAG=IFALSE                                                 AROE 0490
      READ (5,61)X,Y,Z,ISTAT,XX,YY,ZZ,ISTATT,ITYPE                  AROE 0500
      IF (ITYPE .NE. 7) GO TO 51                                    AROE 0510
   60 IF(ISTAT)11,180,11                                            AROE 0520
   11 IF(ISTAT-3)12,180,12                                          AROE 0530
   12 IF(ISTAT-2)70,199,70                                          AROE 0540
   70 IF(IAFLAG-1)13,199,13                                         AROE 0550
   13 MC=M                                                          AROE 0560
   80 M=1                                                           AROE 0570
      IF(ISTAT-2)14,150,14                                          AROE 0580
   14 IF(IBFLAG-1)75,84,75                                          AROE 0590
   75 DO 81 J=1,MC                                                  AROE 0600
      XA(J)=XB(J)                                                   AROE 0610
      YA(J)=YB(J)                                                   AROE 0620
   81 ZA(J)=ZB(J)                                                   AROE 0630
   83 XB(1)=X                                                       AROE 0640
      YB(1)=Y                                                       AROE 0650
      ZB(1)=Z                                                       AROE 0660
      GO TO 30                                                      AROE 0670
   84 IF(IAFLAG)15,85,15                                            AROE 0680
   15 IBFLAG=0                                                      AROE 0690
      GO TO 75                                                      AROE 0700
   85 IAFLAG=1                                                      AROE 0710
```

DECK AROE

```
      GO TO 83                                                          AROE  0720
  150 IAFLAG=0                                                          AROE  0730
      IBFLAG=1                                                          AROE  0740
      N=N+1                                                             AROE  0750
  160 XA(M)=X                                                           AROE  0760
      YA(M)=Y                                                           AROE  0770
      ZA(M)=Z                                                           AROE  0780
      GO TO 30                                                          AROE  0790
  180 M=M+1                                                             AROE  0800
      IF(IAFLAG)16,160,16                                               AROE  0810
   16 XB(M)=X                                                           AROE  0820
      YB(M)=Y                                                           AROE  0830
      ZB(M)=Z                                                           AROE  0840
      IF(ISTAT-3)30,199,30                                              AROE  0850
 199  ML=MC                                                             AROE  0860
      MC=M                                                              AROE  0870
  198 N=N+1                                                             AROE  0880
C                                                                       AROE  0890
C   SET UP BOUNDARY CURVE COORDINATE ARRAYS                             AROE  0900
  157 CONTINUE                                                          AROE  0910
      IF(II-1)40,40,41                                                  AROE  0920
   40 DO 42 I=1,ML                                                      AROE  0930
      XB1(II,I)=XA(I)                                                   AROE  0940
      YB1(II,I)=YA(I)                                                   AROE  0950
   42 ZB1(II,I)=ZA(I)                                                   AROE  0960
      NPTS(II)=ML                                                       AROE  0970
   41 II=II+1                                                           AROE  0980
      DO 43 I=1,MC                                                      AROE  0990
      XB1(II,I)=XB(I)                                                   AROE  1000
      YB1(II,I)=YB(I)                                                   AROE  1010
   43 ZB1(II,I)=ZB(I)                                                   AROE  1020
      NPTS(II)=MC                                                       AROE  1030
      IF(II-4)80,18,18                                                  AROE  1040
   18 CONTINUE                                                          AROE  1050
      IF(ISTAT-3)1,19,19                                                AROE  1060
   19 CONTINUE                                                          AROE  1070
```

```
DECK AROE

C
C
C *******CALCULATE BOUNDARY CURVE CONSTANTS
C  CALCULATE ARC LENGTH S ON BOUNDARY
      NB=1
   22 S=0.0
      K=NPTS(NB)-2
      DO 6 I=2,K
    60S=S+SQRT((XB1(NB,I+1)-XB1(NB,I))**2+(YB1(NB,I+1)-YB1(NB,I))**2+(ZB
     41(NB,I+1)-ZB1(NB,I))**2)
C  CALCULATE TANGENT VECTORS AT THE START OF THE BOUNDARY
      IFLAG1=0
      J1=1
      J2=2
      J3=3
   25 X2X1=XB1(NB,J2)-XB1(NB,J1)
      X3X1=XB1(NB,J3)-XB1(NB,J1)
      X3X2=XB1(NB,J3)-XB1(NB,J2)
      Y2Y1=YB1(NB,J2)-YB1(NB,J1)
      Y3Y1=YB1(NB,J3)-YB1(NB,J1)
      Y3Y2=YB1(NB,J3)-YB1(NB,J2)
      Z2Z1=ZB1(NB,J2)-ZB1(NB,J1)
      Z3Z1=ZB1(NB,J3)-ZB1(NB,J1)
      Z3Z2=ZB1(NB,J3)-ZB1(NB,J2)
      L21=SQRT(X2X1*X2X1+Y2Y1*Y2Y1+Z2Z1*Z2Z1)
      L31=SQRT(X3X1*X3X1+Y3Y1*Y3Y1+Z3Z1*Z3Z1)
      L32=SQRT(X3X2*X3X2+Y3Y2*Y3Y2+Z3Z2*Z3Z2)
      L1=X3X1/L31
      M1=Y3Y1/L31
      N1=Z3Z1/L31
      L2=X2X1/L21
      M2=Y2Y1/L21
      N2=Z2Z1/L21
      LN=-(N1*(L1*N2-L2*N1)+M1*(L1*M2-L2*M1))
      MN=-(N1*(M1*N2-M2*N1)+L1*(L1*M2-L2*M1))
      NN=M1*(M1*N2-M2*N1)+L1*(L1*N2-L2*N1)
```

DECK AROE

```
      COSEP1=(X2X1*X3X1+Y2Y1*Y3Y1+Z2Z1*Z3Z1)/(L21*L31)
      IF(COSEP1-0.999999)32,32,31
   31 EPS1=0.0
      GO TO 35
   32 IF(COSEP1+0.999999)33,34,34
   33 EPS1=0.0
      GO TO 35
   34 EPS1=ARCOS(COSEP1)
   35 COSEP2=(X3X2*X3X1+Y3Y2*Y3Y1+Z3Z2*Z3Z1)/(L32*L31)
      IF(COSEP2-0.999999)37,37,36
   36 EPS2=C.0
      GO TO 44
   37 IF(COSEP2+0.999999)38,39,39
   38 EPS2=0.0
      GO TO 44
   39 EPS2=ARCOS(COSEP2)
   44 DELTA=EPS1+EPS2
      TX=L1*COS(DELTA)+LN*SIN(DELTA)
      TY=M1*COS(DELTA)+MN*SIN(DELTA)
      TZ=N1*COS(DELTA)+MN*SIN(DELTA)
C
C  CALCULATE END POINT DERIVATIVES
      IF(IFLAG1)23,23,24
   23 X1V00=TX*S
      Y1V00=TY*S
      Z1V00=TZ*S
      J1=NPTS(NB)-2
      J2=NPTS(NB)-1
      J3=NPTS(NB)
      IFLAG1=1
      GO TO 25
   24 X1V01=TX*S
      Y1V01=TY*S
      Z1V01=TZ*S
C
C
```

```
DECK AROE

C *********CALCULATE CONSTANTS FOR BOUNDARY CURVE
      D(NB,1)=2.0*(XB1(NB,2)-XB1(NB,J2))+X1V00+X1V01
      D(NB,2)=3.0*(XB1(NB,J2)-XB1(NB,2))-2.0*X1V00-X1V01
      D(NB,3)=X1V00
      D(NB,4)=2.0*(YB1(NB,2)-YB1(NB,J2))+Y1V00+Y1V01
      D(NB,5)=3.0*(YB1(NB,J2)-YB1(NB,2))-2.0*Y1V00-Y1V01
      D(NB,6)=Y1V00
      D(NB,7)=2.0*(ZB1(NB,2)-ZB1(NB,J2))+Z1V00+Z1V01
      D(NB,8)=3.0*(ZB1(NB,J2)-ZB1(NB,2))-2.0*Z1V00-Z1V01
      D(NB,9)=Z1V00
      NB=NB+1
      IF(NB-4)22,22,26
C
C *********CALCULATE PATCH DATA
   26 NOW=NOW/2*2+1
      DELU=1.0/FLOAT(NOU)
      DELW=1.0/FLOAT(NOW)
      NOU=NOU+1
      NOW=NOW+1
      STATT=0
      U=0.0
C
      DO 95 I=1,NOU
      STAT=1
      INU=0
      W=0.0
C
      DO 93 K=1,NOW
C
C
      W3=W**3
      W2=W**2
      U3=U**3
      U2=U**2
C  CALCULATE BLENDING FUNCTIONS
      F1U=3.0*U2-2.0*U3
```

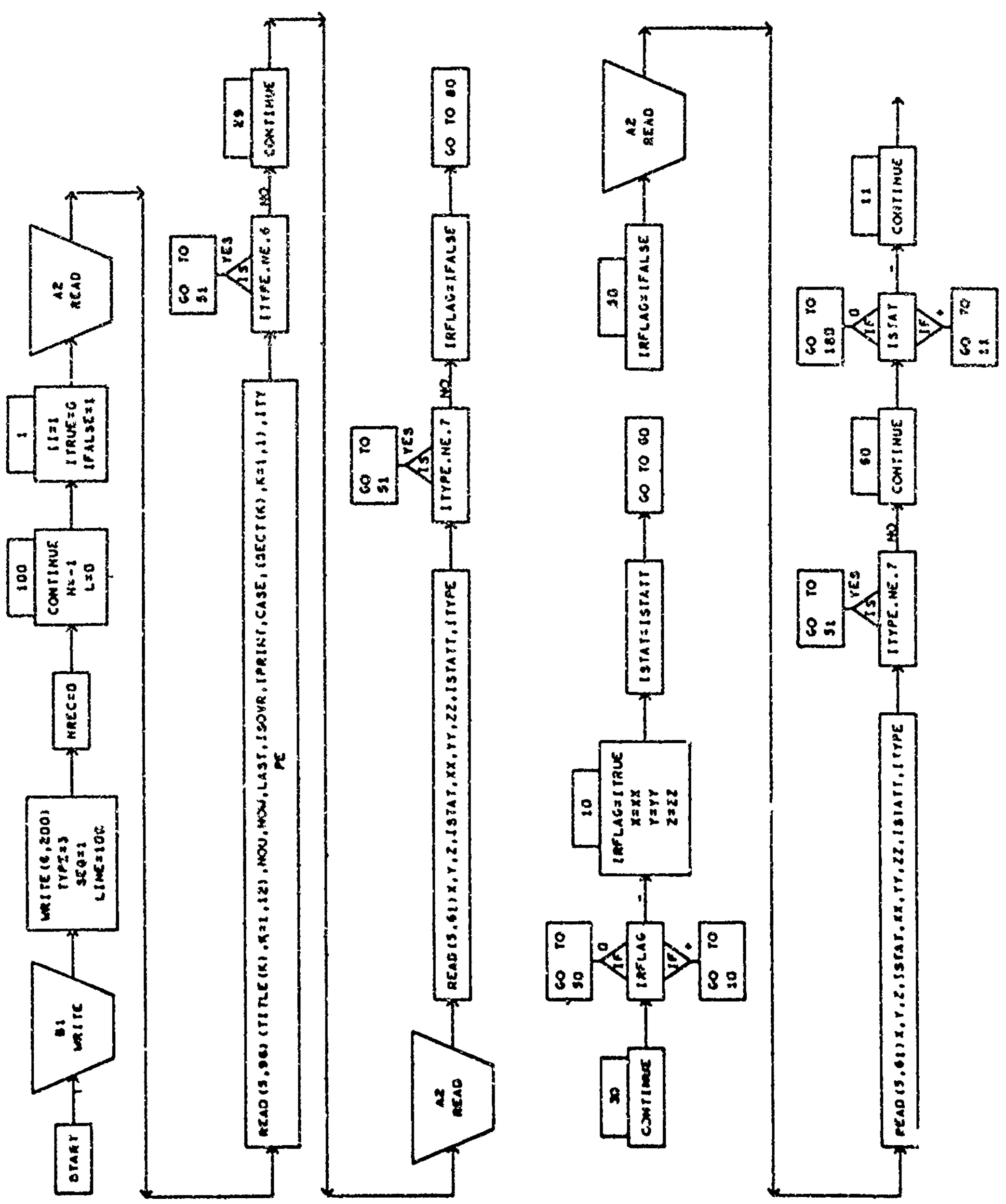
DECK AROE

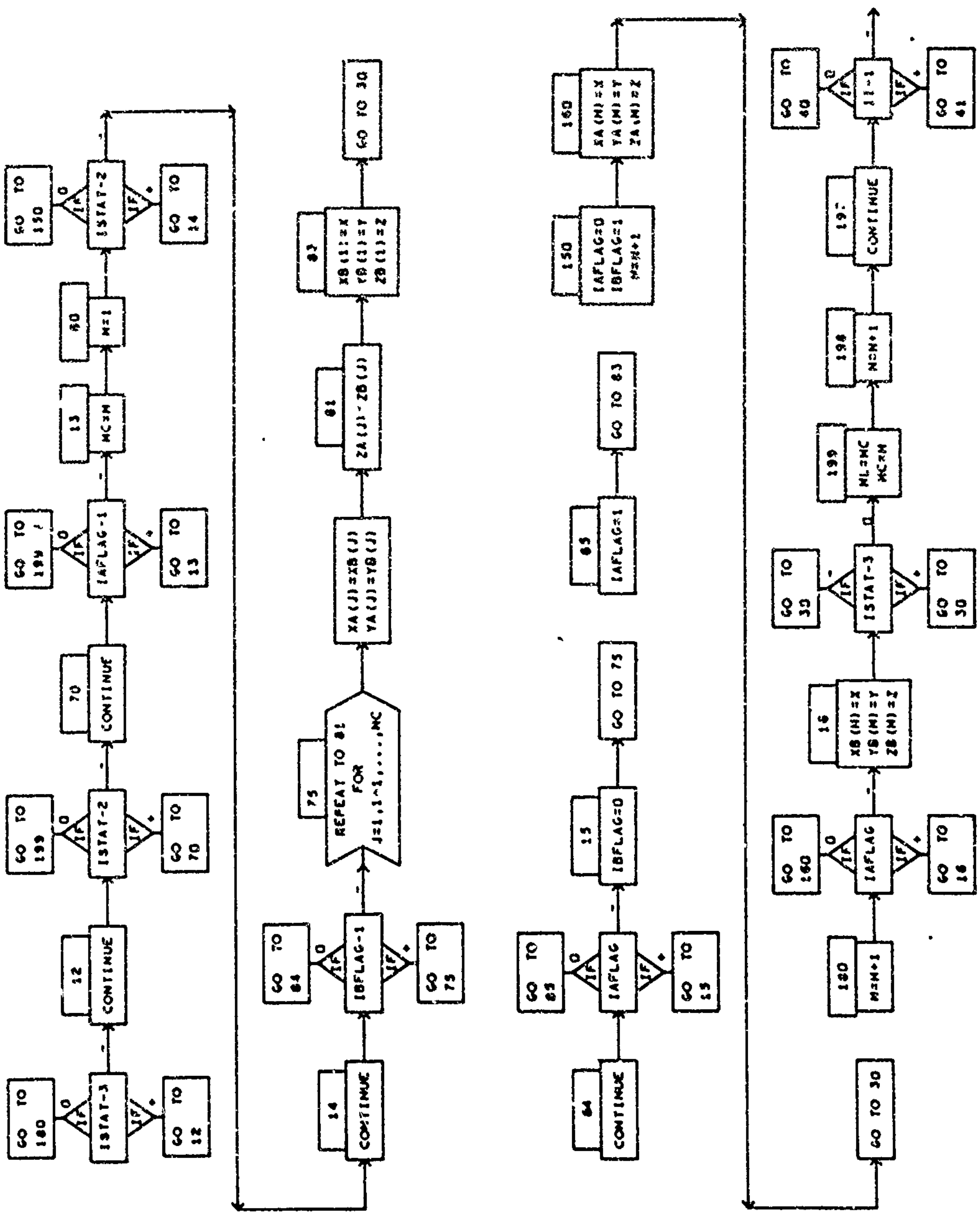
```
      FOU=1.0-3.0*U2+2.0*U3
      F1W=3.0*W2-2.0*W3
      FOW=1.0-3.0*W2+2.0*W3
C
C  CALCULATE POINTS ON BOUNDARY CURVES
      XOW=D(1,1)*W3+D(1,2)*W2+D(1,3)*W+XB1(1,2)
      YOW=D(1,4)*W3+D(1,5)*W2+D(1,6)*W+YB1(1,2)
      ZOW=D(1,7)*W3+D(1,8)*W2+D(1,9)*W+ZB1(1,2)
      X1W=D(2,1)*W3+D(2,2)*W2+D(2,3)*W+XB1(2,2)
      Y1W=D(2,4)*W3+D(2,5)*W2+D(2,6)*W+YB1(2,2)
      Z1W=D(2,7)*W3+D(2,8)*W2+D(2,9)*W+ZB1(2,2)
      XUO=D(3,1)*U3+D(3,2)*U2+D(3,3)*U+XB1(3,2)
      YUO=D(3,4)*U3+D(3,5)*U2+D(3,6)*U+YB1(3,2)
      ZUO=D(3,7)*U3+D(3,8)*U2+D(3,9)*U+ZB1(3,2)
      XU1=D(4,1)*U3+D(4,2)*U2+D(4,3)*U+XB1(4,2)
      YU1=D(4,4)*U3+D(4,5)*U2+D(4,6)*U+YB1(4,2)
      ZU1=D(4,7)*U3+D(4,8)*U2+D(4,9)*U+ZB1(4,2)
      NPT1=NPTS(1)-1
      NPT2=NPTS(2)-1
C
C  CALCULATE POSITION OF A POINT ON THE SURFACE
     OXS = XOW*FOU+X1W*F1U+XUO*FOW+XU1*F1W-XB1(1,2)*FOU*FOW-XB1(1,NPT1)*
     4  FOU*F1W-XB1(2,2)*F1U*FOW-XB1(2,NPT2)*F1U*F1W
     OYS = YOW*FOU+Y1W*F1U+YUO*FOW+YU1*F1W-YB1(1,2)*FOU*FOW-YB1(1,NPT1)*
     4  FOU*F1W-YB1(2,2)*F1U*FOW-YB1(2,NPT2)*F1U*F1W
     OZS = ZOW*FOU+Z1W*F1U+ZUO*FOW+ZU1*F1W-ZB1(1,2)*FOU*FOW-ZB1(1,NPT1)*
     4  FOU*F1W-ZB1(2,2)*F1U*FOW-ZB1(2,NPT2)*F1U*F1W
C
      IF(INU-1)97,94,94
   97 XXS = XS
      YYS = YS
      ZZS = ZS
      INU=1
      GO TO 89
   94 IF (I.EQ.NOU .AND. K.EQ.NOW .AND. LAST.EQ.1) STATT = 3
      IF (I.EQ.NOU .AND. K.EQ.NOW .AND. LAST.EQ.3) STATT = 3
```

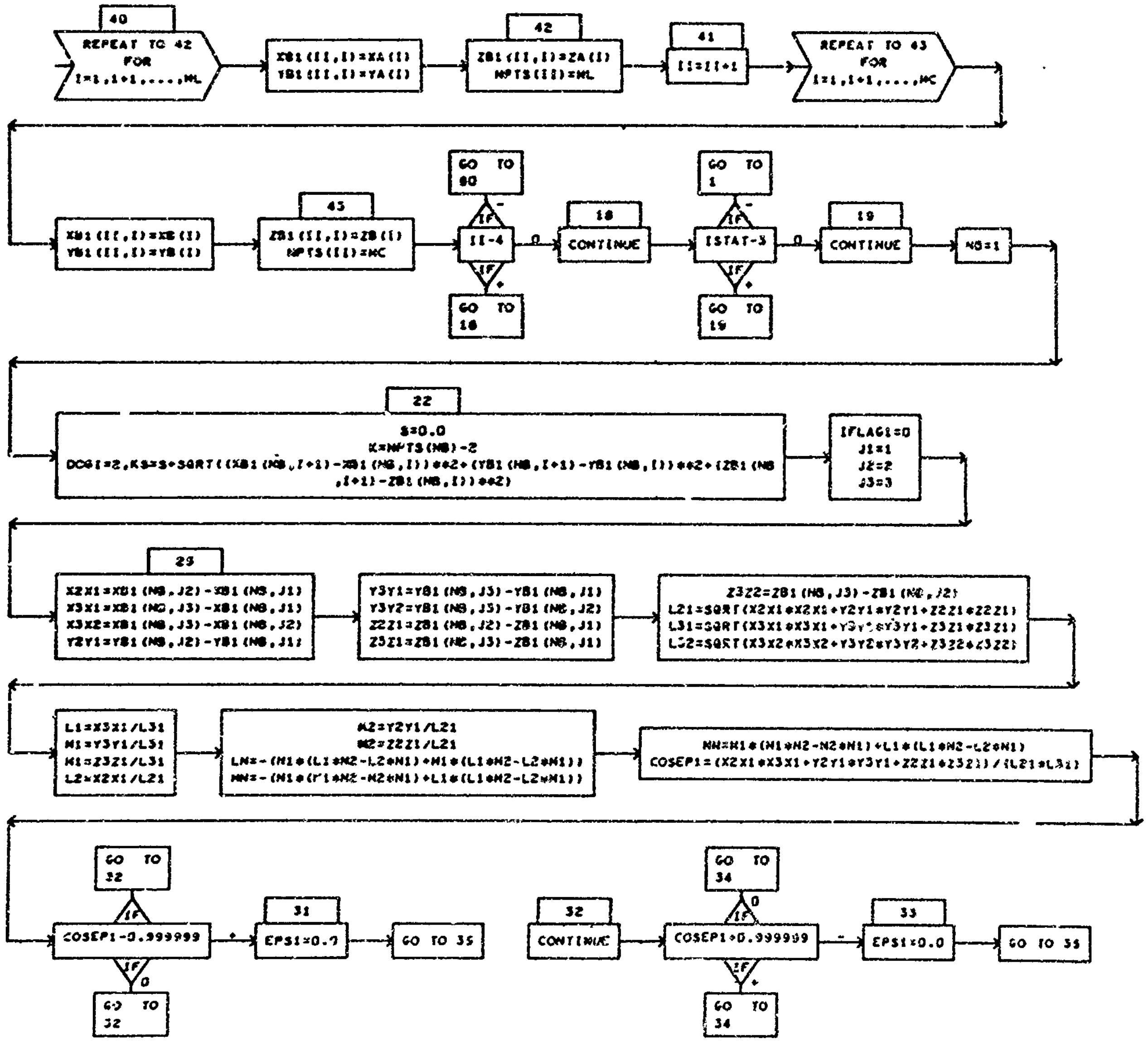
```
DECK AROE

      IF (I.EQ.NOU .AND. K.EQ.NOW .AND. LAST.EQ.4) STATT = 3            AROE  2520
      IF (I.EQ.1 .AND. K.EQ.2 .AND. ISOVR.EQ.0) STAT = 2                AROE  2530
      CALL PUNCH (XXS,YYS,ZZS,STAT,XS,YS,ZS,STATT,SECT,TYPE,LINE,SEQ,   AROE  2540
     1    LAST,IPRINT,NREC)                                             AROE  2550
      INU=0                                                             AROE  2560
      STAT=0                                                            AROE  2570
   89 W=W+DELW                                                          AROE  2580
   93 CONTINUF                                                          AROE  2590
      U=U+DELU                                                          AROE  2600
   95 CONTINUE                                                          AROE  2610
C                                                                       AROE  2620
      IF (STATT .NE. 3) GO TO 100                                       AROE  2630
      WRITE (8,500)                                                     AROE  2640
  500 FORMAT (12H**BLANK CARD,68X)                                      AROE  2650
      END FILE 8                                                        AROE  2660
      BACKSPACE 8                                                       AROE  2670
      BACKSPACE 8                                                       AROE  2680
      IF (LAST .NE. 3) GO TO 1000                                       AROE  2690
      DO 99 I=1,NREC                                                    AROE  2700
   99 BACKSPACE 8                                                       AROE  2710
 1000 RETURN                                                            AROE  2720
   51 EPROR = 1                                                         AROE  2730
      RETURN                                                            AROE  2740
      END                                                               AROE  2750
```

ANALY2

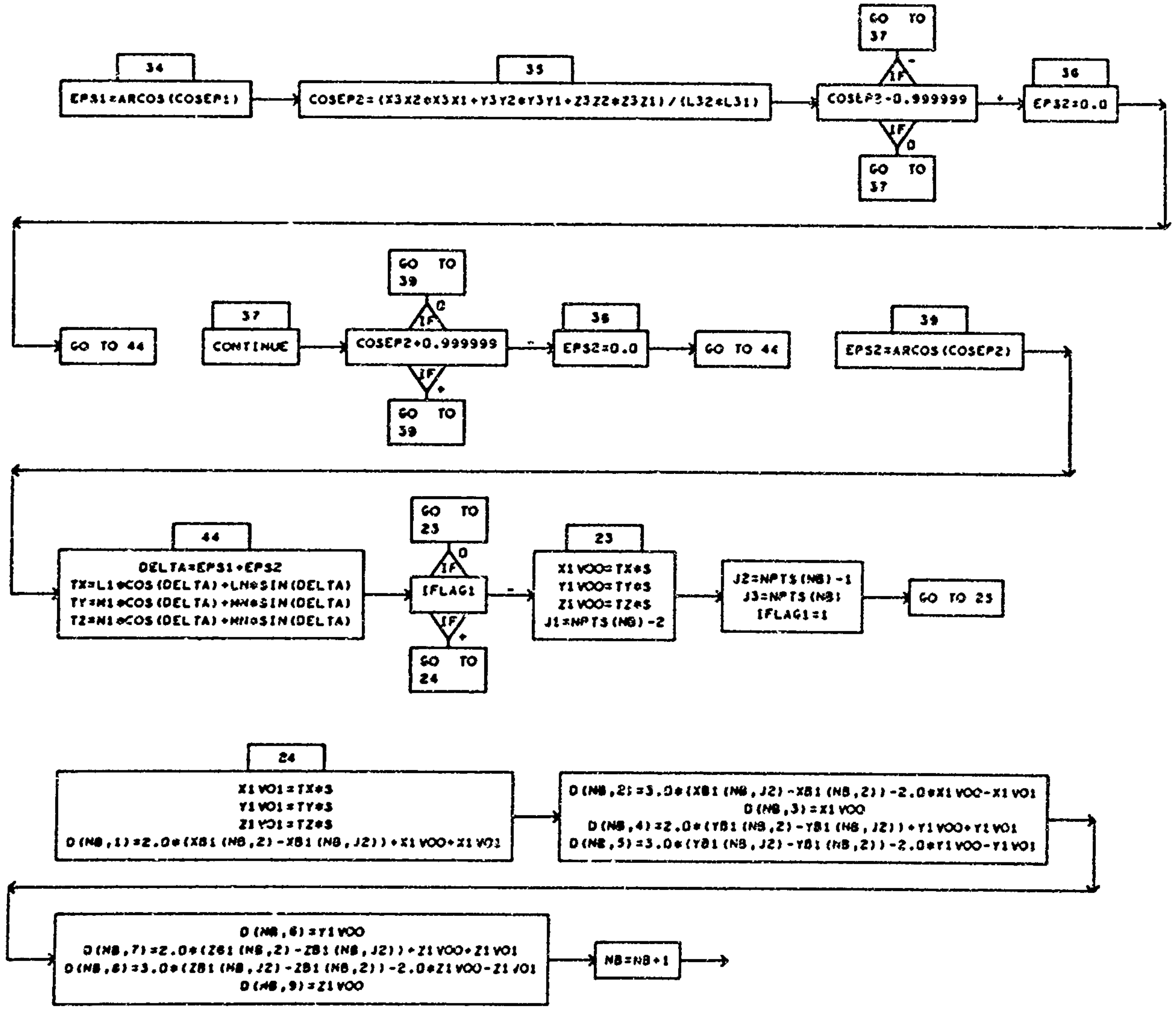
SUBROUTINE ANALY2



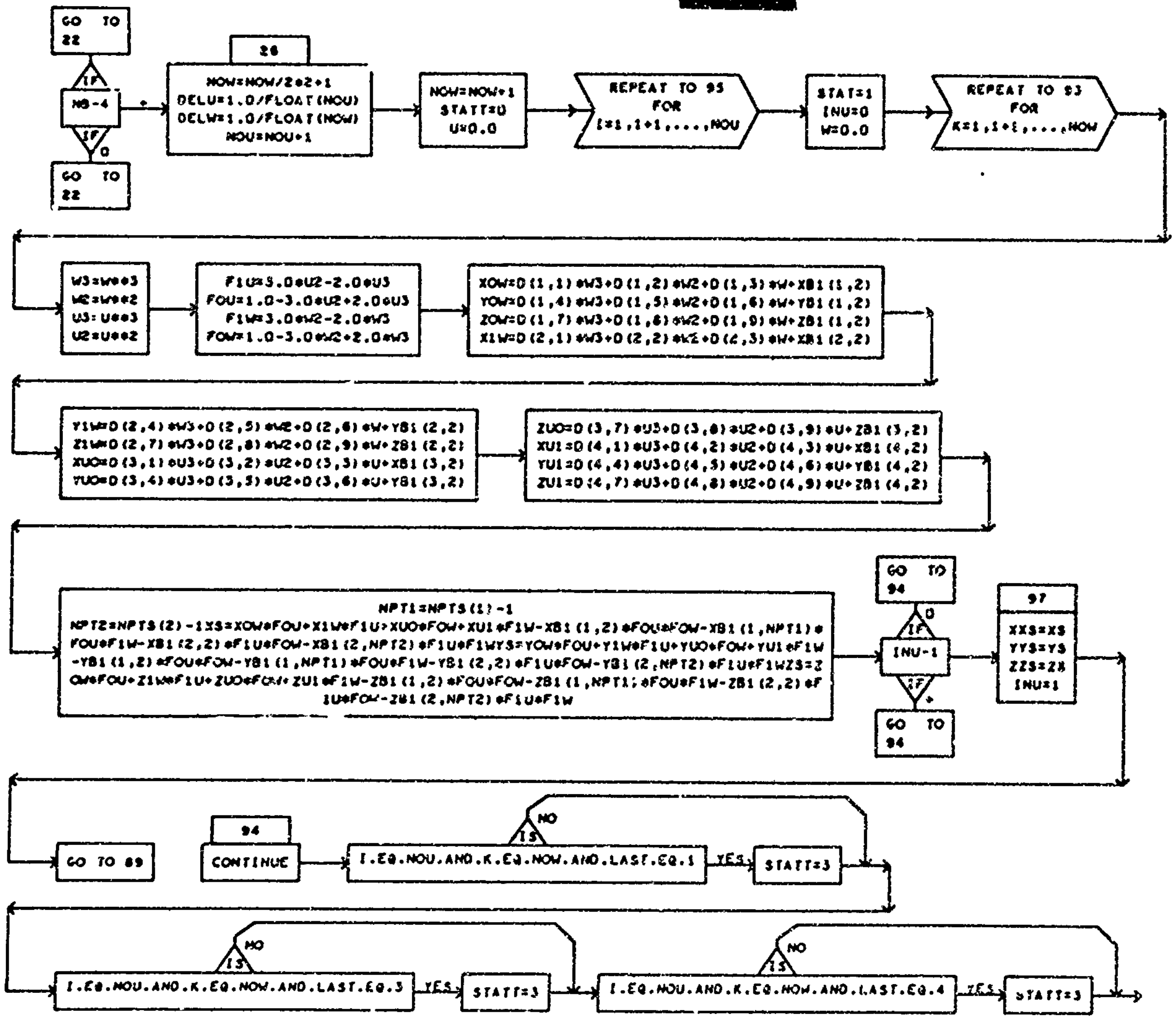




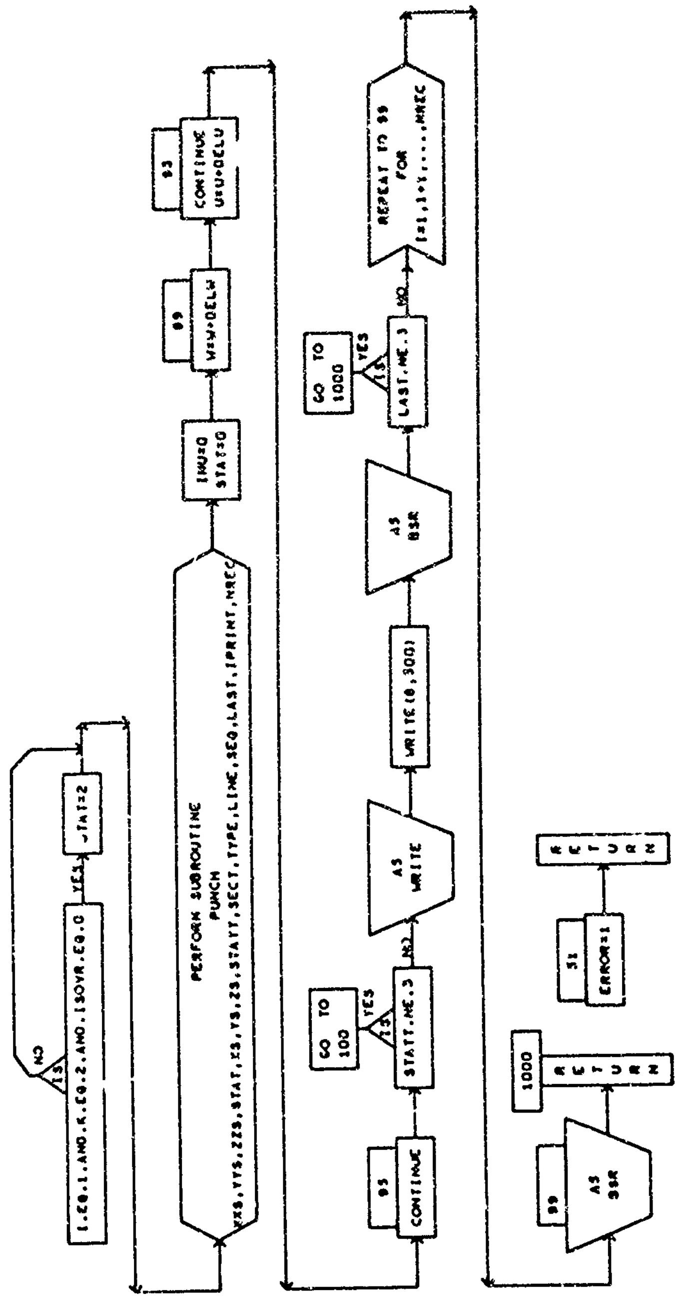
40
REPEAT TO 42 FOR I=1,1+1,....,NL
XB1(II,I)=XA(I)
YB1(II,I)=YA(I)
42
ZB1(II,I)=ZA(I)
NPTS(II)=NL
41
II=II+1
REPEAT TO 43 FOR I=1,1+1,....,NC
XB1(II,I)=XB(I)
YB1(II,I)=YB(I)
43
ZB1(II,I)=ZB(I)
NPTS(II)=NC
IF II-4
GO TO 90
GO TO 18
18
CONTINUE
IF ISTAT-3
GO TO 1
GO TO 19
19
CONTINUE
NB=1
22
S=0.0
K=NPTS(NB)-2
DO OI=2,KS=S+SQRT((XB1(NB,I+1)-XB1(NB,I))**2+(YB1(NB,I+1)-YB1(NB,I))**2+(ZB1(NB,I+1)-ZB1(NB,I))**2)
IFLAG1=0
J1=1
J2=2
J3=3
25
X2X1=XB1(NB,J2)-XB1(NB,J1)
X3X1=XB1(NB,J3)-XB1(NB,J1)
X3X2=XB1(NB,J3)-XB1(NB,J2)
Y2Y1=YB1(NB,J2)-YB1(NB,J1)
Y3Y1=YB1(NB,J3)-YB1(NB,J1)
Y3Y2=YB1(NB,J3)-YB1(NB,J2)
Z2Z1=ZB1(NB,J2)-ZB1(NB,J1)
Z3Z1=ZB1(NB,J3)-ZB1(NB,J1)
Z3Z2=ZB1(NB,J3)-ZB1(NB,J2)
L21=SQRT(X2X1*X2X1+Y2Y1*Y2Y1+Z2Z1*Z2Z1)
L31=SQRT(X3X1*X3X1+Y3Y1*Y3Y1+Z3Z1*Z3Z1)
L32=SQRT(X3X2*X3X2+Y3Y2*Y3Y2+Z3Z2*Z3Z2)
L1=X3X1/L31
M1=Y3Y1/L31
N1=Z3Z1/L31
L2=X2X1/L21
M2=Y2Y1/L21
N2=Z2Z1/L21
LN=-(N1*(L1*N2-L2*N1)+M1*(L1*M2-L2*M1))
MN=-(N1*(M1*N2-M2*N1)+L1*(L1*M2-L2*M1))
NN=M1*(M1*N2-N2*M1)+L1*(L1*N2-L2*N1)
COSEP1=(X2X1*X3X1+Y2Y1*Y3Y1+Z2Z1*Z3Z1)/(L21*L31)
IF COSEP1-0.999999
GO TO 32
GO TO 32
31
EPS1=0.7
GO TO 35
32
CONTINUE
IF COSEP1+0.999999
GO TO 34
GO TO 34
33
EPS1=0.0
GO TO 35


34
EPS1=ARCOS(COSEP1)
35
COSEP2=(X3X2*X3X1+Y3Y2*Y3Y1+Z3Z2*Z3Z1)/(L32*L31)
COSEP2-0.999999
IF
GO TO 37
GO TO 37
36
EPS2=0.0
GO TO 44
37
CONTINUE
COSEP2+0.999999
GO TO 39
GO TO 39
38
EPS2=0.0
GO TO 44
39
EPS2=ARCOS(COSEP2)
44
DELTA=EPS1+EPS2
TX=L1*COS(DELTA)+LN*SIN(DELTA)
TY=M1*COS(DELTA)+MN*SIN(DELTA)
TZ=N1*COS(DELTA)+NN*SIN(DELTA)
IFLAG1
GO TO 23
GO TO 24
23
X1VOO=TX*S
Y1VOO=TY*S
Z1VOO=TZ*S
J1=NPTS(NB)-2
J2=NPTS(NB)-1
J3=NPTS(NB)
IFLAG1=1
GO TO 25
24
X1VO1=TX*S
Y1VO1=TY*S
Z1VO1=TZ*S
D(NB,1)=2.0*(XB1(NB,2)-XB1(NB,J2))+X1VOO+X1VO1
D(NB,2)=3.0*(XB1(NB,J2)-XB1(NB,2))-2.0*X1VOO-X1VO1
D(NB,3)=X1VOO
D(NB,4)=2.0*(YB1(NB,2)-YB1(NB,J2))+Y1VOO+Y1VO1
D(NB,5)=3.0*(YB1(NB,J2)-YB1(NB,2))-2.0*Y1VOO-Y1VO1
D(NB,6)=Y1VOO
D(NB,7)=2.0*(ZB1(NB,2)-ZB1(NB,J2))+Z1VOO+Z1VO1
D(NB,8)=3.0*(ZB1(NB,J2)-ZB1(NB,2))-2.0*Z1VOO-Z1VO1
D(NB,9)=Z1VOO
NB=NB+1

ANALY



GO TO 22
IF
NB-4
IF
GO TO 22
26
NOW=NOW/2*2+1
DELU=1.0/FLOAT(NOU)
DELW=1.0/FLOAT(NOW)
NOU=NOU+1
NOW=NOW+1
STATT=0
U=0.0
REPEAT TO 95
FOR
I=1,1+1,....,NOU
STAT=1
INU=0
W=0.0
REPEAT TO 93
FOR
K=1,1+1,....,NOW
W3=W**3
W2=W**2
U3=U**3
U2=U**2
F1U=3.0*U2-2.0*U3
FOU=1.0-3.0*U2+2.0*U3
F1W=3.0*W2-2.0*W3
FOW=1.0-3.0*W2+2.0*W3
XOW=D(1,1)*W3+D(1,2)*W2+D(1,3)*W+XB1(1,2)
YOW=D(1,4)*W3+D(1,5)*W2+D(1,6)*W+YB1(1,2)
ZOW=D(1,7)*W3+D(1,8)*W2+D(1,9)*W+ZB1(1,2)
X1W=D(2,1)*W3+D(2,2)*W2+D(2,3)*W+XB1(2,2)
Y1W=D(2,4)*W3+D(2,5)*W2+D(2,6)*W+YB1(2,2)
Z1W=D(2,7)*W3+D(2,8)*W2+D(2,9)*W+ZB1(2,2)
XUO=D(3,1)*U3+D(3,2)*U2+D(3,3)*U+XB1(3,2)
YUO=D(3,4)*U3+D(3,5)*U2+D(3,6)*U+YB1(3,2)
ZUO=D(3,7)*U3+D(3,8)*U2+D(3,9)*U+ZB1(3,2)
XU1=D(4,1)*U3+D(4,2)*U2+D(4,3)*U+XB1(4,2)
YU1=D(4,4)*U3+D(4,5)*U2+D(4,6)*U+YB1(4,2)
ZU1=D(4,7)*U3+D(4,8)*U2+D(4,9)*U+ZB1(4,2)
NPT1=NPTS(1)-1
NPT2=NPTS(2)-1XS=XOW*FOU+X1W*F1U+XUO*FOW+XU1*F1W-XB1(1,2)*FOU*FOW-XB1(1,NPT1)*
FOU*F1W-XB1(2,2)*F1U*FOW-XB1(2,NPT2)*F1U*F1WYS=YOW*FOU+Y1W*F1U+YUO*FOW+YU1*F1W
-YB1(1,2)*FOU*FOW-YB1(1,NPT1)*FOU*F1W-YB1(2,2)*F1U*FOW-YB1(2,NPT2)*F1U*F1WZS=Z
OW*FOU+Z1W*F1U+ZUO*FOW+ZU1*F1W-ZB1(1,2)*FOU*FOW-ZB1(1,NPT1)*FOU*F1W-ZB1(2,2)*F
1U*FOW-ZB1(2,NPT2)*F1U*F1W
GO TO 94
IF
INU-1
IF
GO TO 94
97
XXS=XS
YYS=YS
ZZS=ZS
INU=1
GO TO 89
94
CONTINUE
IS
NO
I.EQ.NOU.AND.K.EQ.NOW.AND.LAST.EQ.1
YES
STATT=3
IS
NO
I.EQ.NOU.AND.K.EQ.NOW.AND.LAST.EQ.3
YES
STATT=3
IS
NO
I.EQ.NOU.AND.K.EQ.NOW.AND.LAST.EQ.4
YES
STATT=3


I.EQ.1.AND.K.EQ.2.AND.ISOVR.EQ.0
IS
NO
YES
PERFORM SUBROUTINE
PUNCH
STATT.NE.3
GO TO
100
AS
WRITE
AS
BSR
LAST.NE.3
GO TO
1000
CONTINUE
95
ERROR=1
RETURN
1000
RETURN

```
SYMBOLS USED IN SUBROUTINE ANALY2

CASE     I  C  CASE NUMBER                                 ANALY2
COSEP1   R  U  COSINE OF EPSILON 1                         ANALY2
COSEP2   R  U  COSINE OF EPSILON 2                         ANALY2
D        R  D  CONSTANTS FOR BOUNDARY CURVES               ANALY2
DELTA    R  U  EPSILON 1 + EPSILON 2                       ANALY2
DELU     R  U  DELTA INCREMENT FOR U                       ANALY2
DELW     R  U  DELTA INCREMENT FOR W                       ANALY2
EPS1     R  U  EPSILON 1                                   ANALY2
EPS2     R  U  EPSILON 2                                   ANALY2
ERROR    I  C  ERROR FLAG                                  ANALY2
FOU      R  U  BLENDING FUNCTION VALUE                     ANALY2
FOW      R  U  BLENDING FUNCTION VALUE                     ANALY2
F1U      R  U  BLENDING FUNCTION VALUE                     ANALY2
F1W      R  U  BLENDING FUNCTION VALUE                     ANALY2
I        I  U  DO-LOOP INDEX                               ANALY2
IAFLAG   I  U  INPUT DATA READ CONTROL FLAG                ANALY2
IBFLAG   I  U  INPUT DATA READ CONTROL FLAG                ANALY2
IFALSE   I  U  INPUT DATA READ CONTROL FLAG                ANALY2
IFLAG1   I  U  INPUT DATA READ CONTROL FLAG                ANALY2
II       I  U  BOUNDARY NUMBER                             ANALY2
INU      I  U  FLAG                                        ANALY2
IPRINT   I  U  PRINT FLAG                                  ANALY2
IRFLAG   I  U  INPUT DATA READ CONTROL FLAG                ANALY2
ISOVR    I  U  FIRST POINT STATUS OVERRIDE FLAG            ANALY2
ISTAT    I  U  SURFACE POINT STATUS FLAG                   ANALY2
ISTATT   I  U  SURFACE POINT STATUS FLAG                   ANALY2
ITRUE    I  U  INPUT DATA CONTROL FLAG                     ANALY2
ITYPE    I  U  CARD TYPE NUMBER                            ANALY2
J1       I  U  POINT INDEX                                 ANALY2
J2       I  U  POINT INDEX                                 ANALY2
J3       I  U  POINT INDEX                                 ANALY2
K        I  U  DO-LOOP INDEX                               ANALY2
L        I  U  INDEX                                       ANALY2
LAST     I  U  LAST FLAG                                   ANALY2
LINE     I  U  LINE COUNTER                                ANALY2
LN       R  U  VARIABLE IN TANGENT VECTOR EQUATIONS        ANALY2
```

SYMBOLS USED IN SUBROUTINE ANALY2

L1	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
L2	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
L21	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
L31	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
L32	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
M	I	U	DATA READ IN CONTROL FLAG	ANALY2
MC	I	U	DATA READ IN CONTROL NUMBER	ANALY2
ML	I	U	DATA READ IN CONTROL NUMBER	ANALY2
MN	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
M1	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
M2	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
N	I	U	DATA READ IN COUNTER	ANALY2
NB	I	U	BOUNDARY CURVE NUMBER	ANALY2
NN	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
NCU	I	U	NUMBER OF U INCREMENTS	ANALY2
NCW	I	U	NUMBER OF W INCREMENTS	ANALY2
NPTS	I	D	NUMBER OF BOUNDARY POINTS	ANALY2
NPT1	I	U	BOUNDARY POINT COUNTER	ANALY2
NPT2	I	U	BOUNDARY POINT COUNTER	ANALY2
NREC	I	U	NUMBER OF CARDS WRITTEN ON TAPE 8	ANALY2
N1	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
N2	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
PAGE	I	C	PAGE NUMBER	ANALY2
S	R	U	BOUNDARY LENGTH	ANALY2
SECT	R	D	SECTION IDENTIFICATION	ANALY2
SEQ	I	U	CARD SEQUENCE NUMBER	ANALY2
STAT	I	U	SURFACE POINT STATUS FLAG	ANALY2
STATT	I	U	SURFACE POINT STATUS FLAG	ANALY2
TITLE	R	C	TITLE	ANALY2
TX	R	U	TANGENT VECTOR X-COMPONENT	ANALY2
TY	R	U	TANGENT VECTOR Y-COMPONENT	ANALY2
TYPE	I	U	CARD TYPE NUMBER	ANALY2
TZ	R	U	TANGENT VECTOR Z-COMPONENT	ANALY2
U	R	U	PARAMETRIC VARIABLE, U	ANALY2
U2	R	U	PARAMETRIC VARIABLE U SQUARED	ANALY2
U3	R	U	PARAMETRIC VARIABLE U CUBED	ANALY2

ANALY2

SYMBOLS USED IN SUBROUTINE ANALY2

```
W        R  U  PARAMETRIC VARIABLE, W                         ANALY2
W2       R  U  PARAMETRIC VARIABLE W SQUARED                  ANALY2
W3       R  U  PARAMETRIC VARIABLE W CUBED                    ANALY2
X        R  U  X-COORDINATE                                   ANALY2
XA       R  D  X-COORDINATE                                   ANALY2
XB       R  D  X-COORDINATE                                   ANALY2
XB1      R  D  BOUNDARY CURVE X-COORDINATE ARRAY              ANALY2
X0W      R  U  BOUNDARY CURVE POINT, X(0,W)                   ANALY2
XS       R  U  SURFACE X-COORDINATE POINT                     ANALY2
XU0      R  U  BOUNDARY CURVE POINT, X(U,0)                   ANALY2
XU1      R  U  BOUNDARY CURVE POINT, X(U,1)                   ANALY2
XX       R  U  X-COORDINATE                                   ANALY2
XXS      R  U  X-COORDINATE                                   ANALY2
X1V00    R  U  END POINT DERIVATIVE                           ANALY2
X1V01    R  U  END POINT DERIVATIVE                           ANALY2
X1W      R  U  BOUNDARY CURVE POINT, X(1,W)                   ANALY2
X2X1     R  U  VARIABLE IN TANGENT VECTOR EQUATIONS           ANALY2
X3X1     R  U  VARIABLE IN TANGENT VECTOR EQUATIONS           ANALY2
X3X2     R  U  VARIABLE IN TANGENT VECTOR EQUATIONS           ANALY2
Y        R  U  Y-COORDINATE                                   ANALY2
YA       R  D  Y-COORDINATE                                   ANALY2
YB       R  D  Y-COORDINATE                                   ANALY2
YB1      R  D  BOUNDARY CURVE Y-COORDINATE ARRAY              ANALY2
Y0W      R  U  BOUNDARY CURVE POINT, Y(0,W)                   ANALY2
YS       R  U  SURFACE Y-COORDINATE POINT                     ANALY2
YU0      R  U  BOUNDARY CURVE POINT, Y(U,0)                   ANALY2
YU1      R  U  BOUNDARY CURVE POINT, Y(U,1)                   ANALY2
YY       R  U  Y-COORDINATE                                   ANALY2
YYS      R  U  Y-COORDINATE                                   ANALY2
Y1V00    R  U  END POINT DERIVATIVE                           ANALY2
Y1V01    R  U  END POINT DERIVATIVE                           ANALY2
Y1W      R  U  BOUNDARY CURVE POINT, Y(1,W)                   ANALY2
Y2Y1     R  U  VARIABLE IN TANGENT VECTOR EQUATIONS           ANALY2
Y3Y1     R  U  VARIABLE IN TANGENT VECTOR EQUATIONS           ANALY2
Y3Y2     R  U  VARIABLE IN TANGENT VECTOR EQUATIONS           ANALY2
Z        R  U  Z-COORDINATE                                   ANALY2
```

SYMBOLS USED IN SUBROUTINE ANALY2

ZA	R	D	Z-COORDINATE	ANALY2
ZB	R	D	Z-COORDINATE	ANALY2
ZB1	R	D	BOUNDARY CURVE Z-COORDINATE ARRAY	ANALY2
Z0W	R	U	BOUNDARY CURVE POINT, Z(0,W)	ANALY2
ZS	R	U	SURFACE Z-COORDINATE POINT	ANALY2
ZU0	R	U	BOUNDARY CURVE POINT, Z(U,0)	ANALY2
ZU1	R	U	BOUNDARY CURVE POINT, Z(U,1)	ANALY2
ZZ	R	U	Z-COORDINATE	ANALY2
ZZS	R	U	Z-COORDINATE	ANALY2
Z1V00	R	U	END POINT DERIVATIVE	ANALY2
Z1V01	R	U	END POINT DERIVATIVE	ANALY2
Z1W	R	U	BOUNDARY CURVE COORDINATE, Z(1,W)	ANALY2
Z271	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
Z3Z1	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
Z3Z2	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2

7. SUBROUTINE ANALY3 (DECK AROF)

This is a dummy subroutine provided for future use. This routine may be replaced with a routine similar to either the Ellipse Generation routine or the Parametric Cubic routine.

a. Algorithm

b. Input/Output

c. Error

d. Subroutines Required

e. Argument List

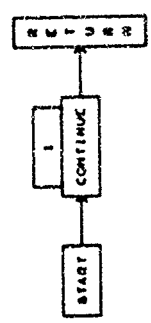
   None

f. Length

   124 bytes

```
DECK AROF

      SURROUTINF ANALY3                                                  AROF  0010
 1    CONTINUE                                                           AROF  0020
      RETURN                                                             AROF  0030
      END                                                                AROF  0040
```

SUBROUTINE ANALYS



8. SUBROUTINE PUNCH (DECK AROG)

This routine writes element data card images on Tape 8.

a. Algorithm

If IPRINT is equal to 1, each card image will be written on output Tape 6. The element data are recorded on Tape 8 in exactly the same form as normal input surface element data (Type 3). Each card is given a sequence number and the number of records written on Tape 8 is furnished to the calling routines.

b. Input/Output

Element data are recorded on Tape 8 and also on the standard output tape if required.

c. Error

None

d. Subroutines Required

None

e. Argument List

(X1. Y1, Z1, NSTAT1, X2, Y2, Z2, NSTAT3, SECT, TYPE, LINE, SEQ, LAST, IPRINT, NREC)

f. Length

1276 bytes

DECK AROG

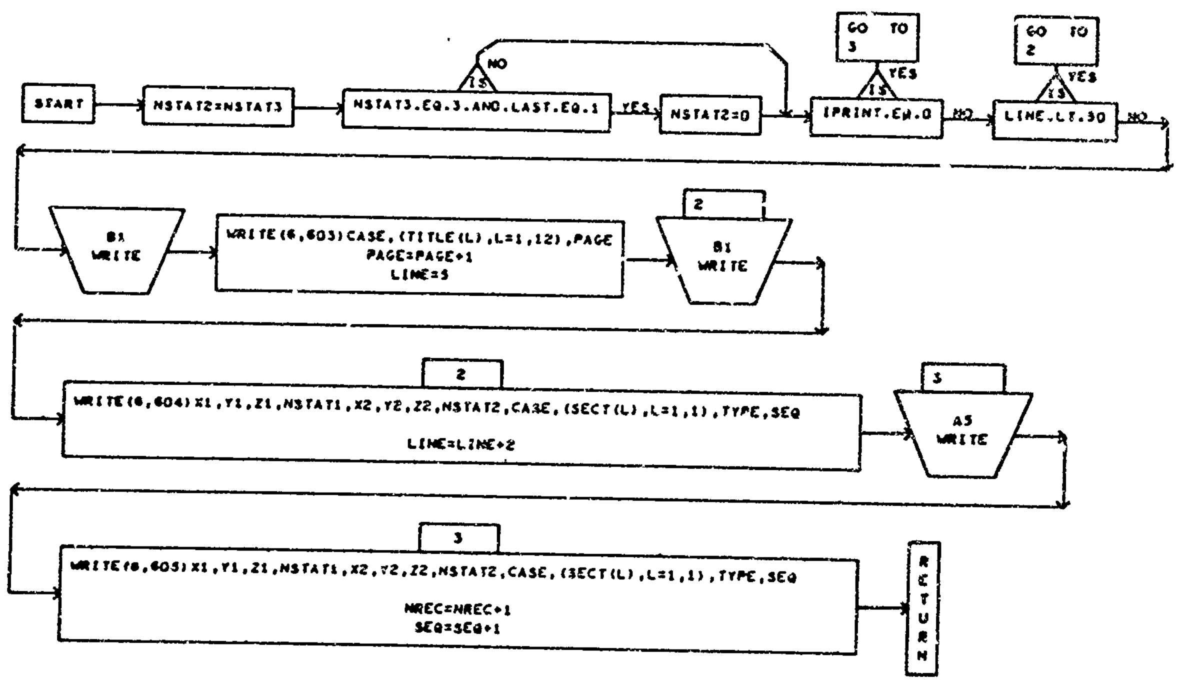
PUNCH

```
      SUBROUTINF PUNCH (X1,Y1,Z1,NSTAT1,X2,Y2,Z2,NSTAT3,SECT,TYPE,
     1      LINE,SEQ,LAST,IPRINT,NREC)
C
C  THIS SUBROUTINE PREPARES VEHICLE GEOMETRY DATA IN THE PROPER FORM
C   FOR USE BY SDATA ROUTINE
C
      DIMENSION  TITLE(15),SECT(1)
C
      COMMON CASE,TITLE,PAGE
      INTEGER  PAGE, SEQ, TYPE
C
      NSTAT2 = NSTAT3
C
C  CHECK IF THIS IS THE LAST POINT OF THE ENTIRE VEHICLE
      IF (NSTAT3.EQ.3 .AND. LAST.EQ.1) NSTAT2 = 0
      IF (IPRINT .EQ. 0) GO TO 3
C
      IF  (LINE .LT. 50 )  GO TO 2
C
C  WRITE PAGE HEADFR FOR STANDARD OUTPUT TAPE
      WRITE (6,603) CASE, (TITLE(L),L=1,12),PAGE
 603  FORMAT (1H1,5X,36HANALYTICALLY GENFRATED ELEMENT DATA ,/
     1  1H0,6H  CASE,I5,17X,12A4,17X,5HPAGE ,I4,/
     2  1H0,5X1HX9X1HY9X1HZ4X1HS5X1HX9X1HY8X1HZ5X1HS10H CASE SECT,
     3  6X3HSEQ )
C
      PAGE = PAGE + 1
      LINE = 5
C
C  WRITE GEOMETRY CARDS ON STANDARD OUTPUT TAPE
    2 WRITE (6,604) X1,Y1,Z1,NSTAT1,X2,Y2,Z2,NSTAT2,CASE,(SECT(L),L=1,1)
     1      ,TYPE,SEQ
  604 FORMAT (1H0,3F10.4,I1,3F10.4,I1,I6,  1A2,1XI1,4HAERO,I4   )
C
      LINE = LINE + 2
```

```
DECK AROG
C                                                                       AROG  0360
C   WRITE GEOMETRY DATA ON GEOMETRY TAPE                                AROG  0370
    3  WRITE (8,605) X1,Y1,Z1,NSTAT1,X2,Y2,Z2,NSTAT2,CASE,(SECT(L),L=1,1) AROG  0380
      1        ,TYPE,SEQ                                                AROG  0390
  605 FORMAT (3F10.4, I1,3F10.4,I1,I6,  1A2,1XI1,4HAERO,I4   )          AROG  0400
C                                                                       AROG  0410
       NREC = NREC + 1                                                  AROG  0420
       SEQ = SEQ + 1                                                    AROG  0430
C                                                                       AROG  0440
       RETURN                                                           AROG  0450
       END                                                              AROG  0460
```

PUNCH

SUBROUTINE PUNCH



SYMBOLS USED IN SUBROUTINE PUNCH

CASE	R	C	CASE NUMBER	PUNCH
IPRINT	I	A	PRINT FLAG	PUNCH
LAST	I	A	LAST FLAG	PUNCH
LINE	I	A	LINE COUNTER	PUNCH
NREC	I	A	NUMBER OF RECORDS ON TAPE 8	PUNCH
NSTAT1	I	A	POINT STATUS FLAG	PUNCH
NSTAT2	I	U	POINT STATUS FLAG	PUNCH
NSTAT3	I	A	POINT STATUS FLAG	PUNCH
PAGE	I	C	PAGE NUMBER	PUNCH
SECT	R	A	SECTION IDENTIFICATION	PUNCH
SEQ	I	A	CARD SEQUENCE NUMBER	PUNCH
TITLE	R	C	TITLE	PUNCH
TYPE	I	A	CARD TYPE NUMBER	PUNCH
X1	R	A	X-COORDINATE	PUNCH
X2	R	A	X-COORDINATE	PUNCH
Y1	R	A	Y-COORDINATE	PUNCH
Y2	R	A	Y-COORDINATE	PUNCH
Z1	R	A	Z-COORDINATE	PUNCH
Z2	R	A	Z-COORDINATE	PUNCH

PUNCH

9. SUBROUTINE CONTRL (DECK AROH)

This routine changes the geometry data for control surfaces to the proper deflected position.

a. Algorithm

The hinge-line coordinate data are determined from the first two items in the NX2 data array. The required transformation angles are then calculated. The element data for the control flap in the undeflected position are read from Tape 4. The rotation matrix and the control deflection matrix are then applied to the geometry data. These data are then rotated back to the original hinge-line centered coordinate system. Hinge moment factor data are calculated and stored in the XCENT2 data array. The new geometry data for the deflected flap are stored on Tape 11.

b. Input/Output

None

c. Error

An error condition occurs when the total number of elements on the control surface is greater than ISIZ (300 for the Mark II program).

d. Subroutines Required

None

e. Argument List

(DELTAE, ISIZ)

f. Length

2376 bytes

DECK AROH

```
      SUBROUTINE CONTRL (DELTAE,ISIZ)
C
C  THIS SUBROUTINE READS CONTROL SURFACE GEOMETRY DATA,
C  ROTATES THE CONTROL SURFACE TO THE REQUIRED NEW POSITION,
C  AND STORES THE NEW GEOMETRY DATA FOR USE BY THE FORCE PROGRAM.
C
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),
     1  YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)
      DIMENSION XPA(4),YPA(4),ZPA(4),XPAD(4),YPAD(4),ZPAD(4),TITLE(15)
C
      COMMON  CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1  AREA2,IN,IM,L,LS
C
      REAL  NX2,NY2,NZ2,NX,NY,NZ,NXD,NYD,NZD,NN,LXY,LYZ
      INTEGER  PAGE,CASE,ERROR
C
      REWIND 4
      REWIND 11
C  SET UP HINGE LINE COORDINATE DATA
      IL = IN(2)
      XHL1 = NX2(1)
      YHL1 = NY2(1)
      ZHL1 = NZ2(1)
      XHL4 = NX2(2)
      YHL4 = NY2(2)
      ZHL4 = NZ2(2)
C
C  CALCULATE TRANSFORMATION ANGLES
C
      LXY=SQRT((XHL1-XHL4)**2+(YHL1-YHL4)**2)
      LYZ=SQRT((XHL1-XHL4)**2+(YHL1-YHL4)**2+(ZHL1-ZHL4)**2)
      IF (LXY .NE. 0.0) GO TO 51
      PSIR = 0.0
      SINPSI = 0.0
      GO TO 56
```

```
DECK AROH
C                                                                          AROH  0360
   51  SINPSI = -(XHL1-XHL4)/LXY                                           ARCH  0370
       IF (ABS(SINPSI) .GT. 1.0) SINPSI = SINPSI / ABS(SINPSI)             AROH  0380
       PSIR = ARSIN(SINPSI)                                                AROH  0390
       IF ((YHL1-YHL4) .GE. 0.0) GO TO 56                                  AROH  0400
       PSIR = 3.141593 - PSIR                                              AROH  0410
   56  SINPHI = (ZHL1-ZHL4)/LYZ                                            AROH  0420
       IF (ABS(SINPHI) .GT. 1.0) SINPHI = SINPHI / ABS(SINPHI)             AROH  0430
       PHIR = ARSIN(SINPHI)                                                AROH  0440
       PSI = PSIR*.5729578E2                                               AROH  0450
       PHI = PHIR*.5729578E2                                               AROH  0460
       COSPSI = COS(PSIR)                                                  AROH  0470
       COSPHI = COS(PHIR)                                                  AROH  0480
       COSDE = COS(DELTAE/.5729578E2)                                      AROH  0490
       SINDE = SIN(DELTAE/.5729578E2)                                      AROH  0500
C                                                                          AROH  0510
C  READ ELEMENT DATA FROM TAPE 4 AND AFTER DEFLECTING THE SURFACE          AROH  0520
C  SAVE THE FINAL GEOMETRY ON TAPE 11 FOR USE BY THE FORCE PROGRAM.        AROH  0530
C                                                                          AROH  0540
       DO 11 J=1,IL                                                        AROH  0550
C                                                                          AROH  0560
       READ (4)LL,N,M,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,YPA,ZPA,XLE      AROH  0570
C                                                                          AROH  0580
       IFLAG=1                                                             AROH  0590
       XP=XCENT-XHL4                                                       AROH  0600
       YP=YCENT-YHL4                                                       AROH  0610
       ZP=ZCENT-ZHL4                                                       AROH  0620
C                                                                          AROH  0630
     4 CONTINUE                                                            AROH  0640
C  APPLY ROTATION MATRIX  E  TO XP, YP, AND ZP                             AROH  0650
       XOP=XP*COSPSI+YP*SINPSI                                             AROH  0660
       YOP=-XP*COSPHI*SINPSI+YP*COSPHI*COSPSI+ZP*SINPHI                    AROH  0670
       ZOP=XP*SINPHI*SINPSI-YP*SINPHI*COSPSI+ZP*COSPHI-                    AROH  0680
C                                                                          AROH  0690
C  APPLY CONTROL DEFLECTION MATRIX                                         AROH  0700
       XOPDE=XOP*COSDE-ZOP*SINDE                                           AROH  0710
```

CONTRL

CONTRL

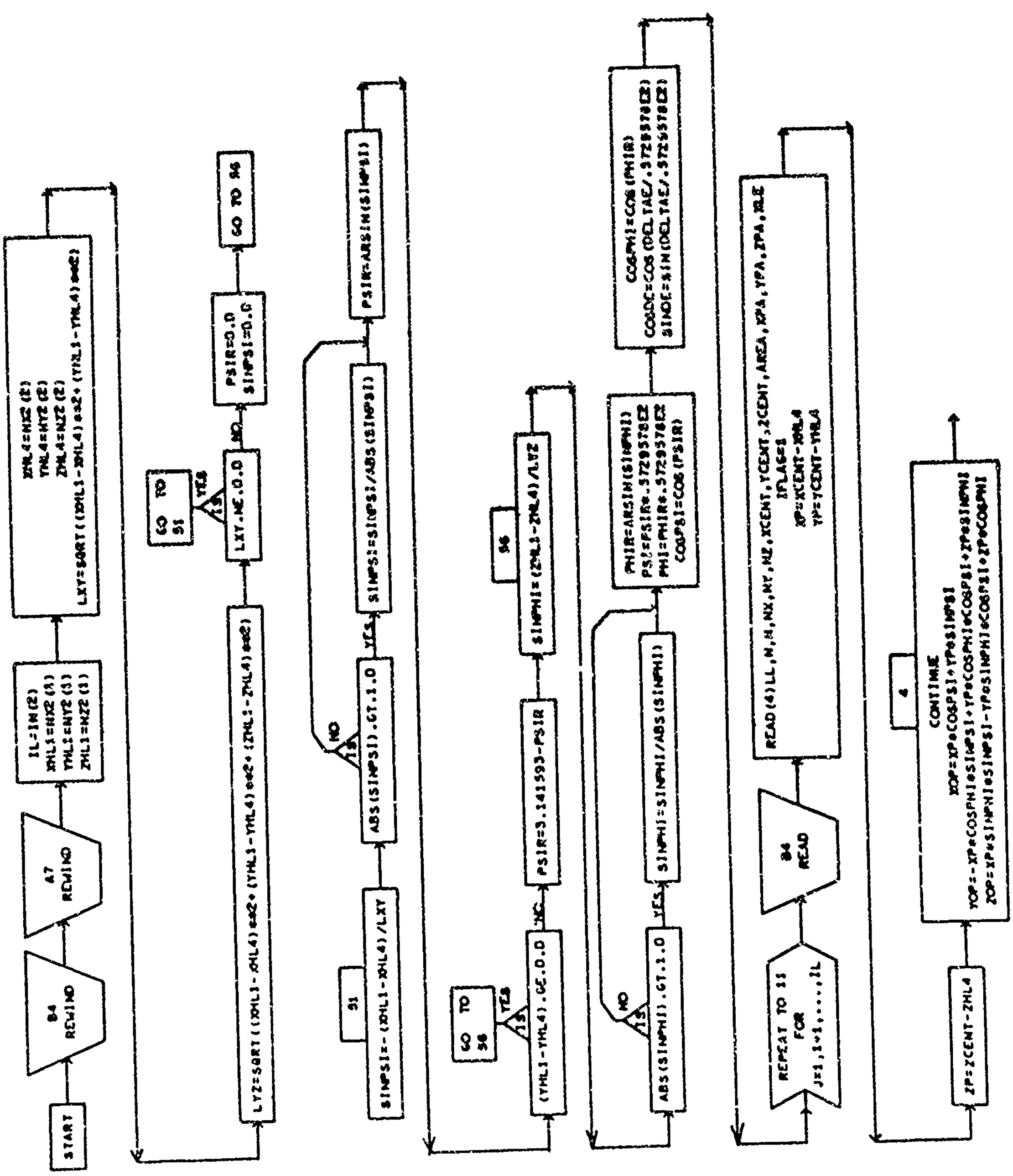
DECK AROH

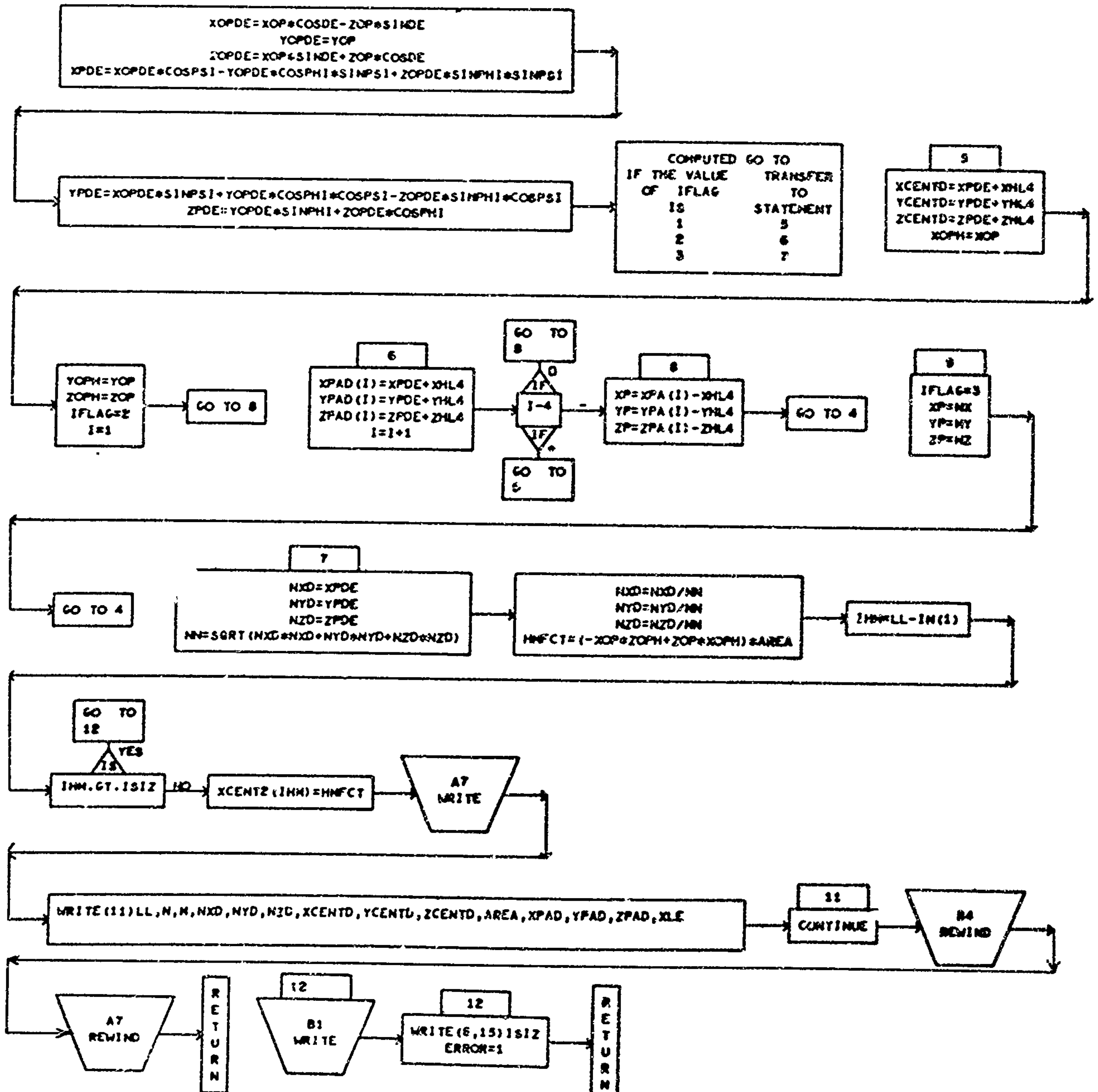
```
      YOPDE=YOP
      ZOPDE=XOP*SINDE+ZOP*COSDE
C
C
C  ROTATE POINT BACK TO ORIGINAL HINGE LINE CENTERED COORDINATE
C  SYSTEM.
C
      XPDE=XOPDE*COSPSI-YOPDE*COSPHI*SINPSI+ZOPDE*SINPHI*SINPSI
      YPDE=XOPDE*SINPSI+YOPDE*COSPHI*COSPSI-ZOPDE*SINPHI*COSPSI
      ZPDE=YOPDE*SINPHI+ZOPDE*COSPHI
C
      GO TO (5,6,7),IFLAG
    5 XCENTD = XPDE + XHL4
      YCENTC = YPDE + YHL4
      ZCENTD = ZPDE + ZHL4
      XOPH=XOP
      ZOPH=ZOP
      IFLAG=2
      I=1
      GO TO 8
C
    6 XPAD(I) = XPDE + XHL4
      YPAD(I) = YPDE + YHL4
      ZPAD(I) = ZPDE + ZHL4
      I=I+1
      IF(I-4)8,8,9
C
    8 XP=XPA(I)-XHL4
      YP=YPA(I)-YHL4
      ZP=ZPA(I)-ZHL4
      GO TO 4
C
    9 IFLAG=3
      XP = NX
      YP = NY
      ZP = NZ
```

```
DECK AROH
      GO TO 4
C
    7 NXD = XPDE
      NYD = YPDE
      NZD = ZPDE
      NN = SQRT(NXD*NXD + NYD*NYD + NZD*NZD)
      NXD = NXD/NN
      NYD = NYD/NN
      NZD = NZD/NN
C
C  CALCULATE HINGE MOMENT FACTOR
      HMFCT = (-XOP*ZOPH + ZOP*XLH) * AREA
      IHM = LL - IN(1)
      IF (IHM .GT. ISIZ) GO TO 12
      XCENT2(IHM) = HMFCT
C
C  SAVE NEW GEOMETRY DATA FOR FORCE PROGRAM
      WRITE (11) LL,N,M,NXD,NYD,NZD,XCENTD,YCENTD,ZCENTD,AREA,XPAD,
     1           YPAD,ZPAD,XLE
C
C
C  END OF ELEMENT DATA READ DO LOOP
   11 CONTINUE
C
      REWIND 4
      REWIND 11
      RETURN
C
   12 WRITE (6,15) ISIZ
   15 FORMAT (1H ,49H***** TOTAL NUMBER OF ELEMENTS ON CONTROL SURFACE
     1 14H CANNOT EXCEED,I4,6H ***** )
      ERROR = 1
      RETURN
      END
```

CONTRL

CONTRL


START
B4
REWIND
A7
REWIND
IL=IN(2)
XHL1=HX2(1)
YHL1=HY2(1)
ZHL1=HZ2(1)
XHL4=HX2(2)
YHL4=HY2(2)
ZHL4=HZ2(2)
LXY=SQRT((XHL1-XHL4)**2+(YHL1-YHL4)**2)
LYZ=SQRT((XHL1-XHL4)**2+(YHL1-YHL4)**2+(ZHL1-ZHL4)**2)
IS
LXY.NE.0.0
YES
GO TO 51
NO
PSIR=0.0
SINPSI=0.0
GO TO 56
51
SINPSI=-(XHL1-XHL4)/LXY
IS
ABS(SINPSI).GT.1.0
YES
NO
SINPSI=SINPSI/ABS(SINPSI)
PSIR=ARSIN(SINPSI)
IS
(YHL1-YHL4).GE.0.0
YES
GO TO 56
NO
PSIR=3.141593-PSIR
56
SINPHI=(ZHL1-ZHL4)/LYZ
IS
ABS(SINPHI).GT.1.0
YES
NO
SINPHI=SINPHI/ABS(SINPHI)
PHIR=ARSIN(SINPHI)
PSI=PSIR*.572957E2
PHI=PHIR*.572957E2
COSPSI=COS(PSIR)
COSPHI=COS(PHIR)
COSDE=COS(DELTAE/.572957E2)
SINDE=SIN(DELTAE/.572957E2)
REPEAT TO 11
FOR
J=1,1+1,....IL
B4
READ
READ(4)LL,N,M,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,YPA,ZPA,XLE
IFLAG=1
XP=XCENT-XHL4
YP=YCENT-YHL4
ZP=ZCENT-ZHL4
4
CONTINUE
XOP=XP*COSPSI+YP*SINPSI
YOP=-XP*COSPHI*SINPSI+YP*COSPHI*COSPSI+ZP*SINPHI
ZOP=XP*SINPHI*SINPSI-YP*SINPHI*COSPSI+ZP*COSPHI


XOPDE=XOP*COSDE-ZOP*SINDE
YOPDE=YOP
ZOPDE=XOP*SINDE+ZOP*COSDE
XPDE=XOPDE*COSPSI-YOPDE*COSPHI*SINPSI+ZOPDE*SINPHI*SINPSI
YPDE=XOPDE*SINPSI+YOPDE*COSPHI*COSPSI-ZOPDE*SINPHI*COSPSI
ZPDE=YOPDE*SINPHI+ZOPDE*COSPHI
COMPUTED GO TO
IF THE VALUE OF IFLAG IS
TRANSFER TO STATEMENT
1 5
2 6
3 7
5
XCENTD=XPDE+XHL4
YCENTD=YPDE+YHL4
ZCENTD=ZPDE+ZHL4
XOPH=XOP
YOPH=YOP
ZOPH=ZOP
IFLAG=2
I=1
GO TO 8
6
XPAD(I)=XPDE+XHL4
YPAD(I)=YPDE+YHL4
ZPAD(I)=ZPDE+ZHL4
I=I+1
GO TO 8
IF
0
I-4
-
IF
+
GO TO 6
8
XP=XPA(I)-XHL4
YP=YPA(I)-YHL4
ZP=ZPA(I)-ZHL4
GO TO 4
9
IFLAG=3
XP=NX
YP=NY
ZP=NZ
GO TO 4
7
NXD=XPDE
NYD=YPDE
NZD=ZPDE
NN=SQRT(NXD*NXD+NYD*NYD+NZD*NZD)
NXD=NXD/NN
NYD=NYD/NN
NZD=NZD/NN
HNFCT=(-XOP*ZOPH+ZOP*XOPH)*AREA
IHH=LL-IN(1)
GO TO 12
YES
IS
IHH.GT.ISIZ
NO
XCENT2(IHH)=HNFCT
A7
WRITE
WRITE(11)LL,N,M,NXD,NYD,NZD,XCENTD,YCENTD,ZCENTD,AREA,XPAD,YPAD,ZPAD,XLE
11
CONTINUE
B4
REWIND
A7
REWIND
RETURN
12
B1
WRITE
12
WRITE(6,15)ISIZ
ERROR=1
RETURN

CONTRL

SYMBOLS USED IN SUBROUTINE CONTRL

AREA	R	U	ELEMENT AREA	CONTRL
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	CONTRL
CASE	I	C	CASE NUMBER	CONTRL
COSDE	R	U	COSINE OF CONTROL DEFLECTION ANGLE	CONTRL
COSPHI	R	U	COSINE OF TRANSFORMATION ANGLE, PHI	CONTRL
COSPSI	R	U	COSINE OF TRANSFORMATION ANGLE, PSI	CONTRL
DELTAE	R	A	CONTROL SURFACE DEFLECTION	CONTRL
ERROR	I	C	ERROR FLAG	CONTRL
HMFCT	R	U	HINGE MOMENT FACTOR	CONTRL
I	I	U	INDEX	CONTRL
IFLAG	I	U	CYCLE FLAG	CONTRL
IHM	I	U	ELEMENT NUMBER INDEX	CONTRL
IL	I	U	NUMBER OF ELEMENTS ON THE FORE SURFACE	CONTRL
IM	I	C	ELEMENT ROW NUMBER ARRAY	CONTRL
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	CONTRL
ISIZ	I	A	NUMBER OF ELEMENTS STORED IN CORE	CONTRL
J	I	U	DO-LOOP INDEX	CONTRL
L	I	C	NUMBER OF ELEMENTS	CONTRL
LL	I	U	ELEMENT NUMBER	CONTRL
LS	I	C	NUMBER OF ELEMENTS	CONTRL
LXY	R	U	HINGE LINE LENGTH IN X-Y PLANE	CONTRL
LYZ	R	U	HINGE LINE LENGTH IN Y-Z PLANE	CONTRL
M	I	U	ELEMENT ROW NUMBER	CONTRL
N	I	U	ELEMENT COLUMN NUMBER	CONTRL
NN	R	U	SURFACE NORMAL LENGTH	CONTRL
NX	R	U	ELEMENT DIRECTION COSINE-X	CONTRL
NXD	R	U	ELEMENT DIRECTION COSINE-X (CONTROL DEFLECTED)	CONTRL
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	CONTRL
NY	R	U	ELEMENT DIRECTION COSINE-Y	CONTRL
NYD	R	U	ELEMENT DIRECTION COSINE-Y (CONTROL DEFLECTED)	CONTRL
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	CONTRL
NZ	R	U	ELEMENT DIRECTION COSINE-Z	CONTRL

SYMBOLS USED IN SUBROUTINE CONTRL

```
NZD      R  U  ELEMENT DIRECTION COSINE-Z (CONTROL DEFLECTED)                    CONTRL
NZ2      R  C  ELEMENT DIRECTION COSINE ARRAY-Z                                  CONTRL
PAGE     I  C  PAGE NUMBER                                                       CONTRL
PHI      R  U  COORDINATE TRANSFORMATION ANGLE, DEGREES                          CONTRL
PHIR     R  U  COORDINATE TRANSFORMATION ANGLE, RADIANS                          CONTRL
PSI      R  U  COORDINATE TRANSFORMATION ANGLE, DEGREES                          CONTRL
PSIR     R  U  COORDINATE TRANSFORMATION ANGLE, RADIANS                          CONTRL
SINDE    R  U  SINE OF CONTROL DEFLECTION ANGLE                                  CONTRL
SINPHI   R  U  SINE OF PHI                                                       CONTRL
SINPSI   R  U  SINE OF PSI                                                       CONTRL
TITLE    R  C  TITLE                                                             CONTRL
XCENT    R  U  QUADRILATERAL ELEMENT CENTROID-X                                  CONTRL
XCENTD   R  U  QUADRILATERAL ELEMENT CENTROID-X (CONTROL DEFLECEDD)              CONTRL
XCENT2   R  C  HINGE MOMENT FACTOR                                               CONTRL
XHL1     R  U  HINGE LINE X-COORDINATE OF POINT 1                                CONTRL
XHL4     R  U  HINGE LINE X-COORDINATE OF POINT 4                                CONTRL
XLE      R  U  DISTANCE FROM LEADING EDGE TO ELEMENT CENTROID                    CONTRL
XOP      R  U  X IN TRANSFORMED SYSTEM                                           CONTRL
XOPDE    R  U  X IN TRANSFORMED SYSTEM (CONTROL DEFLECTED)                       CONTRL
XOPH     R  U  X-COORDINATE                                                      CONTRL
XP       R  U  X-COORDINATE                                                      CONTRL
XPA      R  D  COORDINATES OF ELEMENT CORNER POINTS, X                           CONTRL
XPAD     R  D  COORDINATES OF ELEMENT CORNER POINTS (DEFLECTED)                  CONTRL
XPDE     R  U  X-COORDINATE (DEFLECTED)                                          CONTRL
YCENT    R  U  QUADRILATERAL ELEMENT CENTROID-Y                                  CONTRL
YCENTD   R  U  QUADRILATERAL ELEMENT CENTROID-Y (CONTROL DEFLECTED)              CONTRL
YCENT2   R  C  QUADRILATERAL ELEMENT CENTROID ARRAY-Y                            CONTRL
YHL1     R  U  HINGE LINE Y-COORDINATE OF POINT 1                                CONTRL
YHL4     R  U  HINGE LINE Y-COORDINATE OF POINT 4                                CONTRL
YOP      R  U  Y IN TRANSFORMED SYSTEM                                           CONTRL
YOPDE    R  U  Y IN TRANSFORMED SYSTEM (CONTROL DEFLECTED)                       CONTRL
YOPH     R  U  Y-COORDINATE                                                      CONTRL
YP       R  U  Y-COORDINATE                                                      CONTRL
YPA      R  D  COORDINATES OF ELEMENT CORNER POINTS, Y                           CONTRL
YPAD     R  D  COORDINATES OF ELEMENT CORNER POINTS (DEFLECTED)                  CONTRL
```

CONTRL

SYMBOLS USED IN SUBROUTINE CONTRL

YPDE	R	U	Y-COORDINATE (DEFLECTED)	CONTRL
ZCENT	R	U	QUADRILATERAL ELEMENT CENTROID-Z	CONTRL
ZCENTD	R	U	QUADRILATERAL ELEMENT CENTROID-Z (CONTROL DEFLECTED)	CONTRL
ZCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY, Z	CONTRL
ZHL1	R	U	HINGE LINE Z-COORDINATE OF POINT 1	CONTRL
ZHL4	R	U	HINGE LINE Z-COORDINATE OF POINT 4	CONTRL
ZOP	R	U	Z IN TRANSFORMED SYSTEM	CONTRL
ZOPDE	R	U	Z IN TRANSFORMED SYSTEM (CONTROL DEFLECTED)	CONTRL
ZOPH	R	U	Z-COORDINATE	CONTRL
ZP	R	U	Z-COORDINATE	CONTRL
ZPA	R	D	COORDINATES OF ELEMENT CORNER POINTS, Z	CONTRL
ZPAD	R	D	COORDINATES OF ELEMENT CORNER POINTS (DEFLECTED)	CONTRL
ZPDE	R	U	Z-COORDINATE (DEFLECTED)	CONTRL

10. SUBROUTINE FORCE (DECK AROI)

This routine determines the pressure coefficients on each quadrilateral element, resolves the force in the required body axis system, and sums the contributions of each element to give the vehicle's six aerodynamic coefficients.

a. Algorithm

First the necessary starting constants and conditions are set up. The free-stream properties are then determined either from the Atmosphere subprogram or from the ideal gas equation of NASA TR 1135. The loop for calculating and summing up the forces on each element is then started. The direction cosines of the velocity vector are calculated and the quadrilateral element data are read from core or from tape. Symmetry requirements are then checked and the signs changed if required. The velocity components with vehicle rotation and the impact angle are calculated. If the impact angle is less than zero then the element is in a shadow region; if not, it is in an impact region. The proper force calculation method is then used to determine the pressure coefficient. The force summation method to meet symmetry requirements is selected, the six aerodynamic coefficients calculated, and summation of the coefficients is accomplished. If required, the Skin Friction Subprogram is called for the viscous calculations.

b. Input/Output

If PRINT is equal to 1 the detailed force contributions of each element will be printed.

c. Error

An error will occur when the arccosine of the angle between the velocity vector and the unit normal is greater than 1.

d. Subroutines Required

ATMOS, NEWTPM, COMPR, SHKEXP, EXPAND, FLOSEP, HEADER, SKINFR, BLUNT, CONE

e. Argument List

(ALPHA, BETA, CPSTAG, SREF, SYMFCT, XCG, YCG, ZCG, MACH, SPAN, MAC, J, QRP, ALT, PRINT, CAI, CNI, CYI, CLLI, CLMI, CLNI, ETAC, NS, IMPACT, IPRINT, IFIRST, PSTAG, TSTAG, RENO, ISIZ, ENPM, QQINF, ISHAD, IORIEN, IMPACI, ISHADI, IDERIV, V, IGTYPE, DELTAE, SWEEP, RETRAN, HML, HMR, PFS)

f. Length

13156 bytes

FORCE

DECK AROI

```
      SUBROUTINE FORCE (ALPHA,BETA,CPSTAG,SREF,SYMFCT,XCG,YCG,ZCG,MACH,   AROI  0010
     1  SPAN,MAC,J,QRP,ALT,PRINT,CAI,CNI,CYI,CLLI,CLMI,CLNI,ETAC,NS,       AROI  0020
     2   IMPACT,IPRINT,IFIRST,PSTAG,TSTAG,RENO,ISIZ,ENPM,QQINF,ISHAD,      AROI  0030
     3  IORIEN,IMPACI,ISHADI,IDERIV,V,IGTYPE,DELTAE,SWEEP,RETRAN,           AROI  0040
     4  HML,HMR,PFS,NW,TWALL)                                               AROI  0050
C                                                                           AROI  0060
C************************************************************************** AROI  0070
C**THIS SUBROUTINE CALCULATES THE PRESSURE COEFFICIENT ON EACH SURFACE**   AROI  0080
C**ELEMENT, RESOLVES IT INTO THE REQUIRED DIRECTIONS, AND ADDS UP THE **   AROI  0090
C**CONTRIBUTIONS TO FIND THE TOTAL VEHICLE COEFFICIENTS.               **   AROI  0100
C************************************************************************** AROI  0110
C                                                                           AROI  0120
      DIMENSION  TITLE(15),ALP(20),BET(20),CCA(20),CCY(20),CCN(20),         AROI  0130
     1  CCLL(20),CCLM(20),CCLN(20),CCL(20),CCD(20),CLOD(20),CF(20),         AROI  0140
     2  CPS(20),QQINFS(20),IS(10, 9),SURF(10, 8),ANGLE(3),FS(8),BS(8),      AROI  0150
     3 XPA(4),YPA(4),ZPA(4)                                                 AROI  0160
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),                 AROI  0170
     1  YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)             AROI  0180
C                                                                           AROI  0190
      COMMON  CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,       AROI  0200
     1  AREA2,IN,IM,K,LS,FS,BS,ALP,BET,CCA,CCY,CCN,CCLL,CCLM,CCLN,CCL,      AROI  0210
     2  CCD,CLOD,CF,CPS,QQINFS,IS,SURF,NPRT                                  AROI  0220
C                                                                           AROI  0230
      INTEGER  CASE,PAGE,ERROR,SYMFCT,PRINT,SYMFCO                          AROI  0240
C                                                                           AROI  0250
      REAL  MAC,LOVERD,MACH,NX,NY,NZ                                        AROI  0260
      REAL  NX2,NY2,NZ2                                                     AROI  0270
C                                                                           AROI  0280
      RAD(A) = A /.572957795E+02                                            AROI  0290
C  SET UP NECESSARY STARTING CONSTANTS AND CONDITIONS                       AROI  0300
      REWIND 3                                                              AROI  0310
      REWIND 4                                                              AROI  0320
      REWIND 11                                                             AROI  0330
      AREAT = 0.0                                                           AROI  0340
      NPRT = 14                                                             AROI  0350
```

DECK AROI

```
      NPCK = 13                                                         AROI  0360
      IPRCK = 0                                                         AROI  0370
      SYMFCO = 1                                                        AROI  0380
      ISBP = 0                                                          AROI  0390
      IFSCY = 1                                                         AROI  0400
      DELCPC = 0.0                                                      AROI  0410
      CPNIN = 0.0                                                       AROI  0420
      IMPS = IMPACT                                                     AROI  0430
      ISHS = ISHAD                                                      AROI  0440
      ISPNT = 0                                                         AROI  0450
      DELTAS = DELTAE                                                   AROI  0460
      IREDII = 0                                                        AROI  0470
      CF(J) = 0.0                                                       AROI  0480
      SKIN = 0.0                                                        AROI  0490
      IF (IGTYPE.EQ.1 .AND. DELTAE.NE.0.0) IMPACT = 9                   AROI  0500
      IF (IGTYPE.EQ.1 .AND. DELTAE.NE.0.0) ISHAD = 7                    AROI  0510
      IF (IGTYPE.EQ.1 .AND. IMPACI.EQ.0) IMPACI=3                       AROI  0520
      IF (IGTYPE.EQ.1 .AND. ISHADI.EQ.0) ISHADI = 3                     AROI  0530
      IF (DELTAE .EQ. 0.0) IFSCY = 3                                    AROI  0540
      CA  = 0.0                                                         AROI  0550
      CY  = 0.0                                                         AROI  0560
      CN  = 0.0                                                         AROI  0570
      CLL = 0.0                                                         AROI  0580
      CLM = 0.0                                                         AROI  0590
      HML = 0.0                                                         AROI  0600
      HMR = 0.0                                                         AROI  0610
      CLN = 0.0                                                         AROI  0620
      G = 1.4                                                           AROI  0630
      Q = 0.0                                                           AROI  0640
      R = 0.0                                                           AROI  0650
      P = 0.0                                                           AROI  0660
      ROLL = 0.0                                                        AROI  0670
C   SET UP VEHICLE ROTATION RATES IF REQUIRED                           AROI  0680
      IF (IDERIV.EQ.0.OR.IDERIV.EQ.1.OR.IDERIV.EQ.5) Q = QRP            AROI  0690
      IF (IDERIV.EQ.2.OR.IDERIV.EQ.6) R = QRP                           AROI  0700
      IF (IDERIV .EQ. 3) P = QRP                                        AROI  0710
```

FORCE

FORCE

```
DECK AROI

C                                                                                AROI  0720
C  DETERMINE FREE STREAM PROPERTIES                                              AROI  0730
      IF (PSTAG .GT. 0.0) GO TO 15                                               AROI  0740
C  USE U S 1962 ATMOSPHERE                                                       AROI  0750
      CALL ATMOS (ALT,TFS,PFS,AFS,RHOFS)                                         AROI  0760
      GO TO 16                                                                   AROI  0770
C  USE WIND TUNNEL CONDITIONS  (GAMMA = 1.4) EQ.44,43,29C,26 OF TR 1135         AROI  0780
  15  PFS = PSTAG * (1.0 + (G-1.0)*MACH*MACH/2.)**(-G/(G-1.0))*2116.217          AROI  0790
      TFS = (TSTAG + 459.6) / (1.0 + (G-1.0)*MACH*MACH/2.0)                      AROI  0800
      AFS = 49.021177 * SQRT(TFS)                                                AROI  0810
      RHOFS = PFS / (1716.0 * TFS)                                               AROI  0820
  16  IF(TFS.GE.225.0) VIS =2.27*TFS**1.5/((TFS+198.6)*10.0**8)                  AROI  0830
      IF(TFS.LT.225.0) VIS =0.80382436E-9 * TFS                                  AROI  0840
      V = MACH * AFS                                                             AROI  0850
      RENO = RHOFS * V / VIS                                                     AROI  0860
C                                                                                AROI  0870
C                                                                                AROI  0880
C                                                                                AROI  0890
C                                                                                AROI  0900
C************************************************************************        AROI  0910
C***** START OF LOOP FOR SUMMING UP FORCES ON ELEMENTS                  *****    AROI  0920
C************************************************************************        AROI  0930
  30  L = 0                                                                      AROI  0940
  36  L = L + 1                                                                  AROI  0950
      IF (L.NE.1)GO TO 2                                                         AROI  0960
C                                                                                AROI  0970
C  CALCULATE DIRECTION COSINES OF VELOCITY VECTOR                                AROI  0980
      ALPHAR = RAD(ALPHA)                                                        AROI  0990
      BETAR  = RAD(BETA)                                                         AROI  1000
      ROLLR = RAD(ROLL)                                                          AROI  1010
      PHIR = 0.0                                                                 AROI  1020
      VXI =-V*COS(ALPHAR)*COS(BETAR)                                             AROI  1030
      VYI = V*SIN(BETAR)                                                         AROI  1040
      VZI = V*COS(BETAR)*SIN(ALPHAR)                                             AROI  1050
      VX  = VXI                                                                  AROI  1060
      VY  = VYI                                                                  AROI  1070
```

```
DECK AROI

      VZ  = VZI                                                         AROI  1080
      I4CT = 0                                                          AROI  1090
      IGT = 1                                                           AROI  1100
      CPAVG = 0.0                                                       AROI  1110
      AREAS = 0.0                                                       AROI  1120
C                                                                       AROI  1130
C   SET UP SURFACE ELEMENT DATA                                         AROI  1140
    2 IF (IGTYPE .GT. 0) GO TO 35                                       AROI  1150
      IF (L .GT. ISIZ) GO TO 17                                         AROI  1160
      IF (IMPACT .NE. 16) GO TO 305                                     AROI  1170
      M = IM(L)                                                         AROI  1180
      IF (L .GT. LS) M = 1                                              AROI  1190
      IF (L.EQ.1 .OR. M.GT.1) GO TO 305                                 AROI  1200
C   CALCULATE DATA FOR IMPACT METHOD NUMBER 16                          AROI  1210
      IF (IGT .EQ. 4) GO TO 306                                         AROI  1220
       CPAVG = CPAVG / AREAS                                            AROI  1230
      DELTAR = 0.35                                                     AROI  1240
      L = 0                                                             AROI  1250
  310 L = L + 1                                                         AROI  1260
      EMNS = 1.090909*MACH*SIN(DELTAR) + EXP(-1.090909*MACH*SIN(DELTAR)) AROI  1270
      CP = 2.0*SIN(DELTAR)*SIN(DELTAR) / (1.0-0.25*((EMNS*EMNS+5.0)/    AROI  1280
     1     (6.0*EMNS*EMNS)))                                            AROI  1290
      IF (L .GT. 1) GO TO 312                                           AROI  1300
      DEL1 = DELTAR                                                     AROI  1310
      CP1 = CP                                                          AROI  1320
      DELTAR = DELTAR + 0.05                                            AROI  1330
      GO TO 310                                                         AROI  1340
  312 IF ((ABS(CP-CPAVG).LT.0.0001 ) .OR. L.GT.25) GO TO 313            AROI  1350
      DELT2  = DELTAR - (CP-CPAVG)*(DELTAR-DEL1)/(CP-CP1)               AROI  1360
      IF (DELT2 .GT. 1.5708) DELT2 = 1.57                               AROI  1370
      CP1 = CP                                                          AROI  1380
      DEL1 = DELTAR                                                     AROI  1390
      DELTAR = DELT2                                                    AROI  1400
      GO TO 310                                                         AROI  1410
  313 FINT2 = ((G+1.0)**2 *EMNS*EMNS)/((2.0*G*EMNS*EMNS-(G-1.0))*       AROI  1420
     1         ((G-1.0)*EMNS*EMNS + 2.0))                               AROI  1430
```

FORCE

DECK AROI

```
      EMCONE=SQRT((2.0/(G-1.0))*(TINT2*(1.0+((G-1.)/2.)*MACH*MACH)-1.0))
      L = LSAVE
      IGT = 4
      M = IM(L)
      GO TO 305
  306 IGT = 1
      CPAVG = 0.0
      AREAS = 0.0
  305  NX = NX2(L)
       NY = NY2(L)
       NZ = NZ2(L)
       XCENT = XCENT2(L)
       YCENT = YCENT2(L)
       ZCENT = ZCENT2(L)
       AREA = AREA2(L)
       N = IN(L)
       M = IM(L)
      LL = L
      GO TO 18
C
   17 IF (IMPACT .NE. 16) GO TO 307
      WRITE (6,308) ISIZ
  308 FORMAT (1HO,49H****NUMBER OF ELEMENTS FOR IMPACT PRESSURE METHOD
     1 26H 16 CANNOT BE GREATER THEN,I5)
       ERROR = 1
       GO TO 10
  307 READ (4) LL,N,M,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA
      I4CT = I4CT + 1
      GO TO 18
C
C  READ CONTROL SURFACE GEOMETRY DATA
   35 IF (IGT .EQ. 1)
     1READ ( 3)  LL,N,M,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,YPA,ZPA,XLE
      IREDI1 = 0
      IF (IGTYPE .EQ. 2) GO TO 18
      IF (IGT.EQ.2 .AND. DELTAE.EQ.0.0)
```

DECK AROI

```
      IREAD ( 4)  LL,N,M,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,YPA,ZPA,XLE
       IF (IGT.EQ.2 .AND. DELTAE.NE.0.0)
      IREAD (11)  LL,N,M,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,YPA,ZPA,XLE
       I4CT = I4CT + 1
       IGTS = IGT
       IF(IGTS.EQ.1 .AND. I4CT.EQ.IN(N+10)) IGT = 2
       IF(IGTS.EQ.2 .AND. I4CT.EQ.IN(4)) IGT = 1
       IF (IGT .NE. IGTS) I4CT = 0
C
C  CHECK ON SYMMETRY REQUIREMENTS (CHANGE SIGNS IF REQUIRED)
  18   IF (SYMFCT.EQ.2 .AND. SYMFCO.EQ.2) GO TO 26
       IF (SYMFCT.NE.3) GO TO 27
       GO TO (27,25,26,25), SYMFCO
  25    NZ = -NZ
        ZCENT = -ZCENT
       IF (IGTYPE .EQ. 0) GO TO 22
        DO 21 I=1,4
  21    ZPA(I) = -ZPA(I)
  22   IF (SYMFCO .NE. 4) GO TO 27
  26    NY = -NY
        YCENT = -YCENT
       IF (IGTYPE .EQ. 0) GO TO 27
        DO 29 I=1,4
  29    YPA(I) =-YPA(I)
  27   IF (Q.EQ.0.0 .AND. R.EQ.0.0 .AND. P.EQ.0.0) GO TO 19
C  CALCULATE VELOCITY COMPONENTS WITH VEHICLE ROTATION
       VX = VXI+ (Q*(+ZCENT-ZCG)-R*(+YCENT-YCG))
       VY = VYI+ (R*(+XCENT-XCG)+P*(+ZCENT-ZCG))
       VZ = VZI- (P*(+YCENT-YCG)+Q*(+XCENT-XCG))
C
  19   VLOCAL = SQRT(VX*VX + VY*VY + VZ*VZ)
       IF (IREDI1 .EQ. 1) GO TO 37
C
C  COMPUTE COSINE OF ANGLE BETWEEN UNIT VECTORS
       COSDEL =(-NX*VX - NY*VY - NZ*VZ) / VLOCAL
       IF (COSDEL.GT.-1.0001 .AND. COSDEL.LT.-1.0   ) COSDEL = -1.0
```

FORCE

DECK AROI

```
      IF (COSDEL.GT. 1.0     .AND. COSDEL.LT. 1.0001) COSDEL =  1.0
      IF (COSDEL.GE.-1.0 .AND. COSDEL.LE.1.0) GO TO 20
      WRITE (6,9)
    9 FORMAT (1HO,44H***** FORCE ROUTINE WILL ATTEMPT TO FIND THE
     162H ARCCOSINE OF AN ARGUMENT WHOSE ABSOLUTE VALUE IS GREATER THAN
     210H ONE *****,/1H ,15X30H*** JOB WILL BE TERMINATED ***  )
      ERROR = 1
      GO TO 10
C
C  COMPUTE ANGLE BETWEEN UNIT VECTORS
   20 THETA = ARCOS(COSDEL)
C
C
C
C  COMPUTE NEWTONIAN IMPACT ANGLE
    8 IF (IMPACT.NE.10 .AND. ENPM.NE.0.0) THETA = THETA / ENPM
      DELTAR = 1.57079627 - THETA
      IF (DELTAR.GT.-0.000001 .AND. DELTAR.LT.0.000001) DELTAR = 0.0
C
C  CALCULATE NEWTONIAN IMPACT ANGLE IN DEGREES
      DELTA = DELTAR * .572957795E+02
      IF (IMPACT.EQ.17 .OR. ISHAD.EQ.10) GO TO 314
      IF (IGTYPE .EQ. 2) GO TO 204
C  CHECK TO SEE IF SURFACE IS IN A SHADOW REGION
  314 IF (DELTAR .LT. 0.0) GO TO 5
C
C
C
C
C***********************************************************************
C*********** SELECT IMPACT PRESSURE METHOD      ************************
C***********************************************************************
      GO TO (41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,45,315),IMPACT
C
C  CALCULATE PRESSURE USING MODIFIED NEWTONIAN THEORY
   41 CP = CPSTAG * COSDEL * COSDEL
```

DECK AROI

```
      GO TO 11
C
C  CALCULATE PRESSURE USING NEWTONIAN - PRANDTL-MEYER METHOD
   42 IF (PRINT.EQ.1 .AND. IPRINT.EQ.1) IPRCK = 1
       ANGLE(2) = DELTA
       DO 13 I=1,8
   13   FS(I) = 1.0
        FS(2) = PFS
        FS(6) = MACH
        ISE   = 1
      CALL NEWTPM (ANGLE,EMN,CP,ETAC,IPRCK,MER,CPSTAG,TFS,
     1  PFS,ISE,IFIRST)
      GO TO 11
C
C  CALCULATE PRESSURE USING TANGENT WEDGE
   43 IF (PRINT.EQ.1 .AND. IPRINT.EQ.1) IPRCK= 1
       ANGLE(2) = DELTA
       DO 14 I=1,8
   14   FS(I) = 1.0
        FS(2) = PFS
        FS(6) = MACH
        ISDET = 2
      CALL COMPR (ANGLE,MER,IPRCK,CPSTAG,TFS,PFS,ISDET,IFIRST,CP)
      GO TO 11
C
C  CALCULATE PRESSURE USING TANGENT WEDGE - INFINITE MACH METHOD
   44 EMNS = 1.2*MACH*SIN(DELTAR) + EXP(-0.6 *MACH*SIN(DELTAR))
      CP = 1.6666667 *(EMNS*EMNS - 1.0)/(MACH*MACH)
      GO TO 11
C
C  CALCULATE PRESSURE USING TANGENT CONE EMPIRICAL METHOD
   45 FS(6) = MACH
      ANGLE(1) = DELTA
      CALL CONE (ANGLE,CP,2)
C
      IF (IMPACT .NE. 16) GO TO 11
```

FORCE

DECK AROI

```
      IF (IGT .EQ. 1) LSAVE = L
      IF (IGT .EQ. 1) IGT = 3
      IF (IGT .EQ. 4) GO TO 309
      CPAVG = CPAVG + CP*AREA
      AREAS = AREAS + AREA
      GO TO 36
C  CALCULATE PRESSURE USING MODIFIED TANGENT CONE METHOD
  309 CP = CP - (CP-CPAVG)/EMCONE
      GO TO 11
C
C  CALCULATE PRESSURE USING OSU EMPIRICAL EQUATION (FDL-TDR-64-102)
   46 PPT2 = 0.32 + 0.455*COS(THETA) + 0.195*COS(2.0*THETA) +
     1        0.035*COS(3.0*THETA) - 0.005*COS(4.0*THETA)
      PT2PO = CPSTAG * 0.7*MACH*MACH + 1.0
      CP = (PPT2*PT2PO - 1.0) / (0.7*MACH*MACH)
      GO TO 11
C
C  CALCULATE PRESSURE USING VAN DYKE UNIFIED THEORY
   47 CP = 1.2*DELTAR*DELTAR + SQRT(1.44*DELTAR**4 + 4.0*DELTAR*DELTAR
     1     /(MACH*MACH-1.0))
      GO TO 11
C
C  CALCULATE BLUNT BODY VISCOUS EFFECTS
   48 CP = 0.0
      IVISIN = 1
C
C  THE VISCOUS FORCE COEFFICIENT TAU IS CALCULATED IN
C  SUBROUTINE BLUNT, WHICH ONLY NEEDS TO BE CALLED ONCE
C  FOR EACH SECTION.
C
      IF (L.EQ.1)CALL BLUNT(PFS,MACH,TFS,VIS,RHOFS,ETAC,RENO,TAU,IVISIN)
      SHEAR = TAU*COS(DELTAR)
      GO TO 203
C
C  CALCULATE PRESSURE USING SHOCK-EXPANSION METHOD
   49 CALL SHKEXP (IORIEN,N,M,IPRCK,NX,NY,NZ,DELTA,PFS,MACH,IGTS,
```

```
DECK AROI

      1  DELTAR,IMPACI,CPSTAG,TFS,CP,G,ISHADI,IFIRST,LL,IPRINT,IGTYPE,      AROI
      2  RHOFS,AFS,VIS,V,RENO,O)                                             AROI
       IF (ERROR .EQ. 3) GO TO 10                                            AROI
       GO TO 11                                                              AROI
C                                                                            AROI
C  CALCULATE PRESSURE USING FREE MOLECULAR FLOW                              AROI
   50  FN = CPSTAG                                                           AROI
       FT = ENPM                                                             AROI
       TBTIN = ETAC                                                          AROI
       S = SQRT(G/2.0)*MACH                                                  AROI
       SSIND = S * SIN(DELTAR)                                               AROI
       ERFS = ERF(SSIND)                                                     AROI
       CP =(1.0/(S*S))*((((2.0-FN)/1.7724539*SSIND+FN/2.0*SQRT(TBTIN))*      AROI
      1     EXP(-SSIND*SSIND)) + (((2.0-FN)*(SSIND*SSIND+0.5)+FN/2.0*        AROI
      2     1.7724539*SQRT(TBTIN)*SSIND) * (1.0+ERFS) ))                     AROI
C  CALCULATE SHEAR FORCE COEFFICIENT FOR FREE MOLECULAR FLOW                 AROI
       SHEAR = COS(DELTAR)*FT/(1.7724539*S)* (EXP(-SSIND*SSIND)              AROI
      1         +1.7724539*SSIND*(1.0+ERFS))                                 AROI
       SKIN = SHEAR                                                          AROI
  203  TX = NY*VZ - NZ*VY                                                    AROI
       TY = NZ*VX - NX*VZ                                                    AROI
       TZ = NX*VY - NY*VX                                                    AROI
       SX = TY*NZ - TZ*NY                                                    AROI
       SY = TZ*NX - TX*NZ                                                    AROI
       SZ = TX*NY - TY*NX                                                    AROI
       STOTAL = SQRT(SX*SX + SY*SY + SZ*SZ)                                  AROI
       IF (STOTAL .NE. 0.0) GO TO 200                                        AROI
        SHEARX = 0.0                                                         AROI
        SHEARY = 0.0                                                         AROI
        SHEARZ = 0.0                                                         AROI
       IF (IGTYPE .EQ. 2) GO TO 201                                          AROI
        GO TO 11                                                             AROI
  200  SHEARX = SHEAR * SX / STOTAL                                          AROI
       SHEARY = SHEAR * SY / STOTAL                                          AROI
       SHEARZ = SHEAR * SZ / STOTAL                                          AROI
       IF (IGTYPE .EQ. 2) GO TO 201                                          AROI
```

FORCE

DECK AROI

```
      GO TO 11
C
C  SET PRESSURE COEFFICIENT TO INPUT VALUE
   51 CP = CPSTAG
      GO TO 11
C
C
C  CALCULATE PRESSURE USING HANKEY FLAT SURFACE EMPIRICAL CORRELATION
   52 IF (DELTA .GE. 10.0) GO TO 56
      HANKEY= (0.195+0.222594/MACH**0.3-0.4)*DELTA + 4.0
      GO TO 57
   56 HANKEY= 1.95 + C.3925/(MACH**0.3*TAN(DELTAR))
   57 CP = HANKEY* COSDEL*COSDEL
      GO TO 11
C
C  CALCULATE PRESSURE USING DELTA WING CORRELATICN (SMYTH)
   53 DELDLW = DELTAR
      IF (DELDLW .LT. 0.01745) DELDLW = 0.01745
      EMNS = MACH * SIN(DELDLW)
      EMNS = 1.09*EMNS + EXP(-0.49 *EMNS)
      CP = 1.0/(MACH*MACH) * 1.66667*(EMNS*EMNS-1.0)
      GO TO 11
C
C  CALCULATE PRESSURE USING DAHLEM-BUCK RELATICNSHIP
   54 IF (DELTAR .EQ. 0.0) DELTAR = 0.00001
      CP = 1.0 / (ABS(SIN(4.0*DELTAR)))**0.75 + 1.0
      IF (CP .GT. 5.0) CP = 5.0
      IF (CP.LT.2.C .OR. (DELTA.GT.22.5)) CP = 2.0
      CP = CP * COSDEL*COSDEL
      GO TO 11
C
C  CALCULATE PRESSURE USING BLAST WAVE ANALYSIS
   55 IF (CPSTAG .GT. 0.5) GO TO 300
      CP = (0.067*MACH*MACH*ETAC/(ENPM-XCENT) + 0.44 )
     1    / (G/2.0*MACH*MACH)
      GO TO 11
```

DECK AROI

```
 300  CP = (0.121*MACH*MACH*ETAC/(ENPM-XCENT)**.667 + 0.56 )
     1    /(G/2.0*MACH*MACH)
      GO TO 11
C
C*********************************************************************
C****************  SELECT SHADOW PRESSURE METHOD   ********************
C*********************************************************************
   5  GO TO (61,62,63,64,65,66,49,67,50,315),ISHAD
C
C  CALCULATE PRESSURE IN SHADOW REGIONS
C
C  CALCULATE PRESSURE USING CP=0 IN SHADOW REGIONS
  61  CP = 0.0
      GO TO 11
C
C  CALCULATE PRESSURE USING NEWTONIAN - PRANDTL-MEYER
  62  IF (PRINT.EQ.1 .AND. IPRINT.EQ.1) IPRCK = 1
      ANGLE(2) = DELTA
      DO 23 I=1,8
  23   FS(I) = 1.0
       FS(2) = PFS
       FS(6) = MACH
       ISE   = 1
      CALL NEWTPM (ANGLE,EMN,CP,ETAC,IPRCK,MER,CPSTAG,TFS,
     1   PFS,ISE,IFIRST )
      GO TO 11
C
C  CALCULATE PRESSURE USING PRANDTL-MEYER EXPANSICN FROM FREE STREAM
  63  ANGLE(2) = ABS(DELTA)
      DO 24 I=1,8
  24   FS(I) = 1.0
       FS(2) = PFS
       FS(6) = MACH
       ISDET = 2
      CALL EXPAND (ANGLE,MER,IPRCK,ISDET,CP)
       GO TO 11
```

FORCE

```
DECK AROI

C                                                                        AROI  4320
C  CALCULATE PRESSURE USING OSU EMPIRICAL EQUATION                       AROI  4330
  64  PPT2 = 0.32 + 0.455*COS(THETA) + 0.195*COS(2.0*THETA) +            AROI  4340
     1      0.035*COS(3.0*THETA) - 0.005*COS(4.0*THETA)                  AROI  4350
      PT2PO = CPSTAG * 0.7*MACH*MACH + 1.0                               AROI  4360
      CP = (PPT2*PT2PO - 1.0) / (0.7*MACH*MACH)                          AROI  4370
      GO TO 11                                                           AROI  4380
C                                                                        AROI  4390
C  CALCULATE PRESSURE USING VAN DYKE UNIFIED THEORY                      AROI  4400
  65  CP =(1.42857/(MACH*MACH-1.0))* ((1.0-0.2*SQRT(MACH*MACH-1.0)*      AROI  4410
     1    ABS(DELTAR))**7 - 1.0)                                         AROI  4420
      IF (CP.LT. (-1./(MACH*MACH))) CP = -1.0/(MACH*MACH)                AROI  4430
      GO TO 11                                                           AROI  4440
C                                                                        AROI  4450
C  CALCULATE PRESSURE USING BASE PRESSURE RELATIONSHIP (CP --1/M**2)     AROI  4460
  66  CP = - 1.0 / (MACH*MACH)                                           AROI  4470
      GO TO 11                                                           AROI  4480
C                                                                        AROI  4490
C  SET PRESSURE COEFFICIENT TO INPUT VALUE                               AROI  4500
  67  CP = ETAC                                                          AROI  4510
      GO TO 11                                                           AROI  4520
C                                                                        AROI  4530
C  DETERMINE INDUCED PRESSURE INCREMENT                                  AROI  4540
 315  NS = L                                                             AROI  4550
      IGTYPE = 17                                                        AROI  4560
      CALL SKINFR (ALPHA,CA,SREF,SHEAR,NS,ALT,MACH,CPSTAG,TFS,PFS,AFS,   AROI  4570
     1  RHOFS,IFIRST,VIS,ISIZ,CN,IGTYPE,DELTA)                           AROI  4580
      IF (SURF(NS,1) .GT. 0.00001) AREA = SURF(NS,1)                     AROI  4590
      CP = SHEAR / (G/2.0 * MACH*MACH) * QQINF                           AROI  4600
      GO TO 201                                                          AROI  4610
  11  CONTINUE                                                           AROI  4620
      IF (IMPACT.EQ.16 .AND. IGT.EQ.3) GO TO 36                          AROI  4630
C                                                                        AROI  4640
C ***  GO TO FLOW SEPARATION SUBROUTINE IF REQUIRED  ****                AROI  4650
      IF (IGTYPE.EQ.1 .AND. DELTAS.NE.0.0)                               AROI  4660
     1 CALL FLOSEP (IGT,LL,N,M,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,      AROI  4670
```

DECK AROI

```
     2      YPA,ZPA,IGTYPE,IORIEN,IPRCK,DELTA,PFS,MACH,DELTAR,IMPACI,
     3     CPSTAG,TFS,CP,G,ISHADI,IFIRST,ISBP,IFSCY,IGTS,IMPS,
     4     ISHS,L,XLE,DELTAE,SWEEP,CPNIN,DELCPC,XATACH,XSEPP,ISPNT,
     5     IMPACT,ISHAD,RETRAN,NW,TWALL,IREDI1)
      IF (ERROR .NE. 0) GO TO 10
      IF (IREDI1 .EQ. 1) GO TO 18
C
C  CORRECT CP FOR LOCAL Q
  37  CP = CP * QQINF * VLOCAL*VLOCAL / (V*V)
C
C  CALCULATE SKIN FRICTION IF REQUIRED
 204  IF (IGTYPE .NE. 2) GO TO 201
      NS = L
      CALL SKINFR (ALPHA,CA,SREF,SHEAR,NS,ALT,MACH,CPSTAG,TFS,PFS,AFS,
     1  RHOFS,IFIRST,VIS,ISIZ,CN,IGTYPE,DELTA)
      IF (SURF(NS,1) .GT. 0.00001) AREA = SURF(NS,1)
      SHEAR = SHEAR * QQINF
      CP = 0.0
      GO TO 203
C
C*********************************************************************
C******* PRESSURE CALCULATION PART OF PROGRAM HAS BEEN COMPLETED *******
C*********************************************************************
C
 201  IF (ISBP .EQ. 1) GO TO 4
C
C
C  SELECT FORCE METHOD TO MEET SYMMETRY REQUIREMENTS
      IF (SYMFCT .EQ. 1) GO TO 6
      IF (R.NE.0.0 .OR. P.NE.0.0) GO TO 6
      IF (BETA.EQ.0.0 .AND. ROLL.EQ.0.0) GO TO 28
C
C  CALCULATE SIX-COMPONENT FORCE CONTRIBUTIONS OF ELEMENT
   6  DELCA = NX    * (CP*AREA/SREF)
      DELCY = NY    * (CP*AREA/SREF)
      DELCN =-NZ    * (CP*AREA/SREF)
```

FORCE

DECK AROI

```
      IF (IMPACT.EQ.17 .OR. ISHAD.EQ.10) GO TO 110                     AROI  5040
      IF (IGTYPE .EQ. 2) GO TO 113                                     AROI  5050
      IF (IMPACT.EQ.8 .AND. DELTAR.GE.0.0) GO TO 113                   AROI  5060
      IF ((IMPACT.EQ.10.AND.DELTAR.GE.0.0) .OR.                        AROI  5070
     1    (ISHAD.EQ.9.AND.DELTAR.LT.0.0)) GO TO 113                    AROI  5080
      GO TO 110                                                        AROI  5090
  113 DELCA = DELCA - SHEARX *(AREA/SREF)                              AROI  5100
      DELCY = DELCY - SHEARY *(AREA/SREF)                              AROI  5110
      DELCN = DELCN + SHEARZ *(AREA/SREF)                              AROI  5120
 110  DELCLL= DELCY * (ZCENT - ZCG)/SPAN                               AROI  5130
     1        +DELCN * (YCENT - YCG)/SPAN                              AROI  5140
      DELCLM= DELCN * (XCENT - XCG)/MAC                                AROI  5150
     1        +DELCA * (ZCENT - ZCG)/MAC                               AROI  5160
      DELCLN= DELCY * (XCENT - XCG)/SPAN                               AROI  5170
     1        -DELCA * (YCENT - YCG)/SPAN                              AROI  5180
      GO TO 7                                                          AROI  5190
  28  DELCA = NX    * (CP*AREA/SREF) * 2.0                             AROI  5200
      DELCY =  0.0                                                     AROI  5210
      DELCN =-NZ    * (CP*AREA/SREF) * 2.0                             AROI  5220
      IF (IMPACT.EQ.17 .OR.ISHAD.EQ.10) GO TO 111                      AROI  5230
      IF (IGTYPE .EQ. 2) GO TO 114                                     AROI  5240
      IF (IMPACT.EQ.8 .AND. DELTAR.GE.0.0) GO TO 114                   AROI  5250
      IF ((IMPACT.EQ.10.AND.DELTAR.GE.0.0) .OR.                        AROI  5260
     1    (ISHAD.EQ.9.AND.DELTAR.LT.0.0)) GO TO 114                    AROI  5270
        GO TO 111                                                      AROI  5280
 114  DELCA = DELCA - SHEARX * 2.0*(AREA/SREF)                         AROI  5290
      DELCN = DELCN + SHEARZ * 2.0*(AREA/SREF)                         AROI  5300
 111  DELCLL = 0.0                                                     AROI  5310
      DELCLM=(DELCN * (XCENT - XCG) / MAC                              AROI  5320
     1        +DELCA * (ZCENT - ZCG) / MAC )                           AROI  5330
      DELCLN=  0.0                                                     AROI  5340
C                                                                      AROI  5350
C   SUM UP SIX-COMPONENT FORCE CONTRIBUTIONS                           AROI  5360
    7 CA = CA + DELCA                                                  AROI  5370
      CY = CY + DELCY                                                  AROI  5380
      CN = CN + DELCN                                                  AROI  5390
```

DECK AROI

```
      CLL = CLL + DELCLL                                                AROI  5400
      CLM = CLM + DELCLM                                                AROI  5410
      CLN = CLN + DELCLN                                                AROI  5420
      AREAT = AREAT + AREA                                              AROI  5430
      IF ((IGTYPE.NE.1 .AND. IGTYPE.NE.3) .OR. ((IGTYPE.EQ.1 .OR.       AROI  5440
     1  IGTYPE.EQ.3) .AND. (IGTS.EQ.1))) GO TO 70                       AROI  5450
      IHM = LL - IN(1)                                                  AROI  5460
      HMFCT = XCENT2(IHM)                                               AROI  5470
      DELTHM = HMFCT * CP * G/2.0 *MACH*MACH * PF5                      AROI  5480
      IF (SYMFCO .EQ. 1) HML = HML + DELTHM                             AROI  5490
      IF (SYMFCO .EQ. 2) HMR = HMR + DELTHM                             AROI  5500
   70 IF (PRINT .EQ. 0) GO TO 4                                         AROI  5510
C                                                                       AROI  5520
C  CHECK IF THIS IS A SKIN FRICTION SURFACE.                            AROI  5530
      IF (NS.EQ.0 .OR. IMPACT.EQ.17 .OR. ISHAD.EQ.10) GO TO 71          AROI  5540
C                                                                       AROI  5550
C  SKIN FRICTION SURFACE.  CHECK IF TEMP. ITERATIONS TO BE PRINTED.     AROI  5560
      IF (IS(L,9).EQ.1) GO TO 72                                        AROI  5570
C                                                                       AROI  5580
C  CHECK IF DETAIL AND/OR LOCAL CF DATA TO BE PRINTED.                  AROI  5590
      IF ((IS(L,7).EQ.0).AND.(IS(L,9).EQ.0)) GO TO 71                   AROI  5600
C                                                                       AROI  5610
C  DETAIL AND/OR LOCAL CF DATA TO BE PRINTED.  WRITE ELEMENT DATA       AROI  5620
C  HEADING FOR FIRST SURFACE PER PAGE ONLY.                             AROI  5630
      IF (NPRT - 26) 72,72,73                                           AROI  5640
   72 NPRT = NPRT + 11                                                  AROI  5650
      GO TO 74                                                          AROI  5660
   73 NPRT = NPRT + 3                                                   AROI  5670
      GO TO 12                                                          AROI  5680
C                                                                       AROI  5690
C  CHECK TO PRINT HEADER AT TOP OF PAGE                                 AROI  5700
   71 IF (NPRT.GE.NPCK) GO TO 3                                         AROI  5710
       NPRT = NPRT + 1                                                  AROI  5720
       GO TO 12                                                         AROI  5730
    3 NPRT = 0                                                          AROI  5740
      CALL HEADER                                                       AROI  5750
```

DECK ARO1

```
   74 CONTINUE                                                          ARO1  5760
      WRITE (6,102) MACH,ALT,SREF,SPAN,IMPACT,IMPACI,                   ARO1  5770
     1  XCG,YCG,ZCG,MAC,ISHAO,ISHADI                                    ARO1  5780
  102 FORMAT (1H0,20HELEMENT DATA    MACH=F7.3,7H  ALT =F8.0,9H  S REF =  ARO1  5790
     1 F8.1,8H  SPAN =F7.1,10H  IMPACT =I3,10H  IMPACI =I3,/1H ,        ARO1  5800
     2 15X5HXCG =F7.1,7H  YCG =F7.1,10H    ZCG  =F7.1,4X5HMAC =F7.1,    ARO1  5810
     3 10H   ISHAD =I3,10H  ISHADI =I3)                                 ARO1  5820
      WRITE (6,100) ALPHA,BETA,CPSTAG,ETAC,DELTAS,IDERIV,Q,R,P          ARO1  5830
  100 FORMAT (1H ,5X17HANGLE OF ATTACK =F6.2,3X11HYAW ANGLE =F6.2,      ARO1  5840
     1  3X3HK =      F8.5,3X6HETAC =F8.4,3X,9HDELTA E =F6.2,/1H ,       ARO1  5850
     2  5X8HIDERIV =I3,3X3HQ =E12.5,3X,3HR =E12.5,3X3HP =E12.5,/        ARO1  5860
     3  1H0,2X,1HL,6X,6HDEL CA,8X                                       ARO1  5870
     4  6HDEL CY,8X6HDEL CN,7X7HDEL CLL,7X7HDEL CLM,7X7HDEL CLN,7X2HCF, ARO1  5880
     5  13X4HAREA,/1H ,11X2HCA,12X2HCY,12X2HCN,11X3HCLL,11X3HCLM,       ARO1  5890
     6  12X3HCLN,8X5HDELTA )                                            ARO1  5900
                                                                        ARO1  5910
C  PRINT ELEMENT DATA                                                   ARO1  5920
   12 WRITE (6,101)LL,DELCA,DELCY,DELCN,DELCLL,DELCLM,DELCLN,CP,AREA,   ARO1  5930
     1  CA,CY,CN,CLL,CLM,CLN,DELTA                                      ARO1  5940
  101 FORMAT (1H0,I4,8E14.5,/1H ,4X7E14.5  )                            ARO1  5950
      IF (IGTYPE .NE. 1 .AND. IGTYPE .NE. 3) GO TO 4                    ARO1  5960
      WRITE (6,103) DELCPC,CPNIN,HML,HMR                                ARO1  5970
  103 FORMAT (1H ,7X,18HDELTA CP CONTROL =F12.5,19H  FORCE METHOD CP =  ARO1  5980
     1  E12.5,13H  H.M. (+Y) =E12.5,13H  H.M. (-Y) =E12.5)              ARO1  5990
      IF (ISPNT .EQ. 1) WRITE (6,104) XSEPP                             ARO1  6000
  104 FORMAT (1H ,4X,34H******* FLOW HAS SEPARATED AT   X =E12.5)       ARO1  6010
      IF (ISPNT .EQ. 2) WRITE (6,105) XATACH                            ARO1  6020
  105 FORMAT (1H ,4X,34H******* FLOW HAS ATTACHED AT    X =E12.5)       ARO1  6030
      NPCK = 9                                                          ARO1  6040
C                                                                       ARO1  6050
C  END OF MAJOR LOOP TO SUM UP ELEMENT FORCES                           ARO1  6060
    4 IF (L .LT. LS) GO TO 36                                           ARO1  6070
      IF (IGTYPE.EQ.1 .AND. IFSCY.NE.3) GO TO 30                        ARO1  6080
C                                                                       ARO1  6090
C  SYMMETRY RE-CYCLE CONTROL                                            ARO1  6100
      IF (SYMFCT.EQ.1) GO TO 33                                         ARO1  6110
```

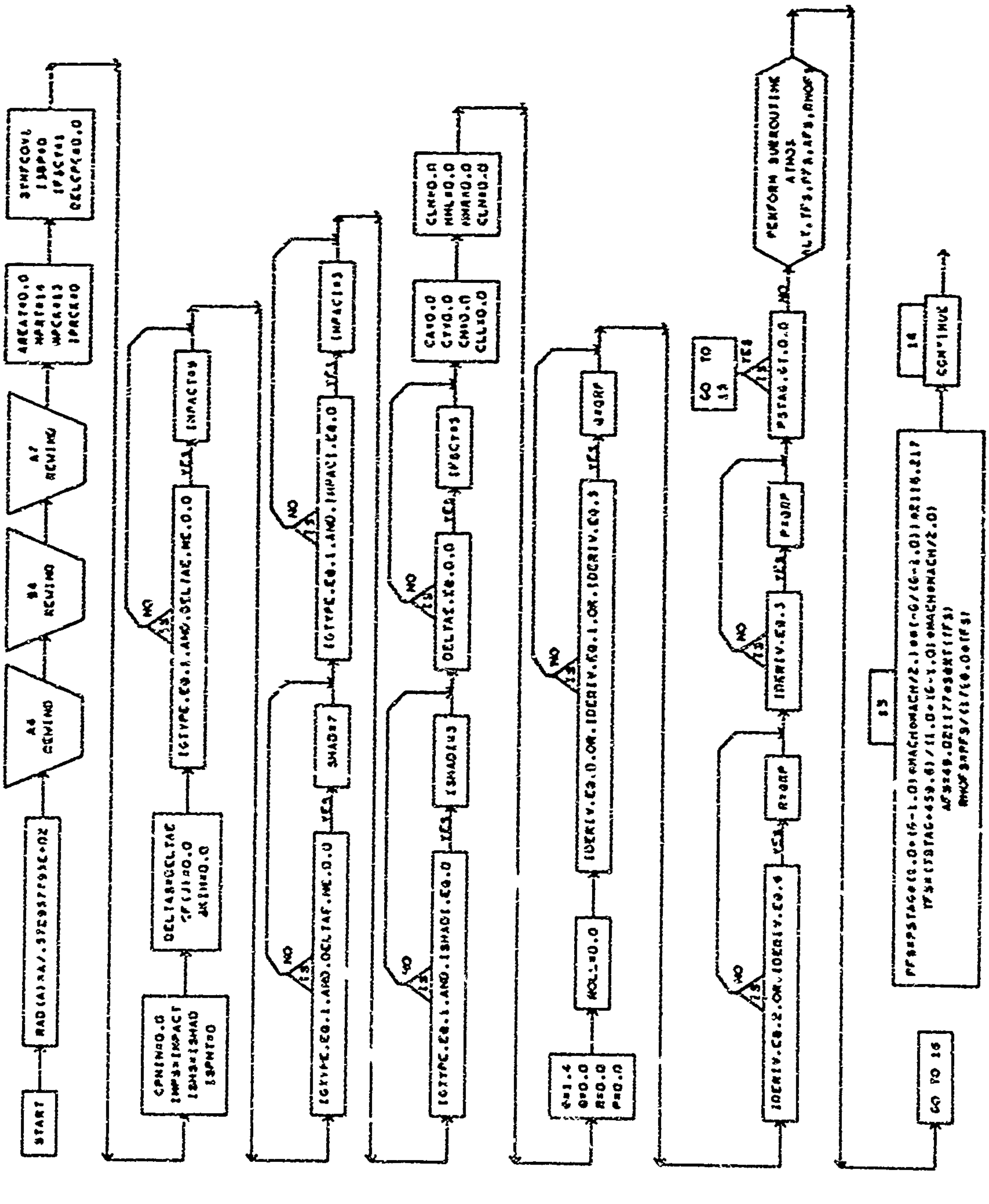
```
DECK AROI

      IF (SYMFCT.EQ.3 .AND. SYMFCO.EQ.1) GO TO 31                        AROI  6120
      IF (R.NE.0.0 .OR. P.NE.0.0) GO TO 31                               AROI  6130
      IF (BETA.EQ.0.0 .AND. ROLL.EQ.0.0) GO TO 33                        AROI  6140
   31 IF (SYMFCT.EQ.2 .AND. SYMFCO.EQ.2) GO TO 33                        AROI  6150
      IF (SYMFCO.EQ.4) GO TO 33                                          AROI  6160
C  SET UP FOR RE-CYCLE                                                   AROI  6170
      SYMFCO = SYMFCO + 1                                                AROI  6180
      IF (IGTYPE .EQ. 0) GO TO 39                                        AROI  6190
      IF (IGTYPE .NE. 1) GO TO 40                                        AROI  6200
      IF (DELTAS .NE. 0.0) IFSCY = 1                                     AROI  6210
       IMPACT = 9                                                        AROI  6220
       ISHAD  = 7                                                        AROI  6230
   40 REWIND 3                                                           AROI  6240
      REWIND 4                                                           AROI  6250
      REWIND 11                                                          AROI  6260
      GO TO 30                                                           AROI  6270
   39 IF (LS.LE.ISIZ) GO TO 30                                           AROI  6280
      DO 32 L=1,14CT                                                     AROI  6290
   32 BACKSPACE 4                                                        AROI  6300
      GO TO 30                                                           AROI  6310
   33 CONTINUE                                                           AROI  6320
C                                                                        AROI  6330
C  CALL SKIN FRICTION ROUTINE                                            AROI  6340
      IF (NS.EQ.0 .OR. IGTYPE.EQ.2) GO TO 112                            AROI  6350
      CALL SKINFR (ALPHA,CA,SREF,SKIN,NS,ALT,MACH,CPSTAG,TFS,PFS,AFS,    AROI  6360
     1  RHOFS,IFIRST,VIS,ISIZ,CN,IGTYPE,DELTA)                           AROI  6370
C                                                                        AROI  6380
C                                                                        AROI  6390
C                                                                        AROI  6400
C  SET UP ARRAYS OF DATA TO BE PRINTED                                   AROI  6410
  112 ALP(J)   = ALPHA                                                   AROI  6420
      BET(J)   = BETA                                                    AROI  6430
      CCA(J)    = CA + CAI                                               AROI  6440
      CCY(J)    = CY + CYI                                               AROI  6450
      CCN(J)    = CN + CNI                                               AROI  6460
      CCLL(J)   = CLL+ CLLI                                              AROI  6470
```

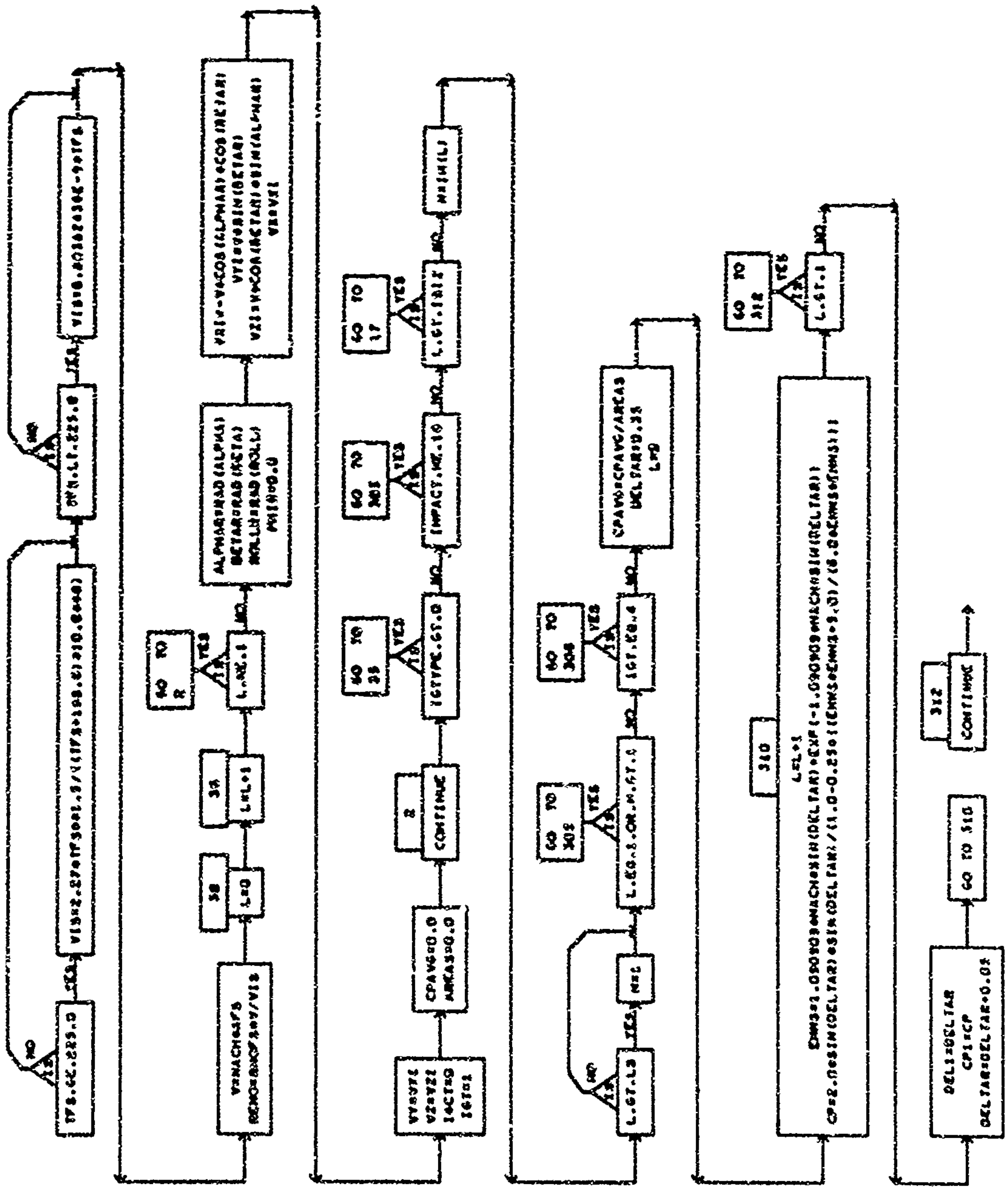
FORCE

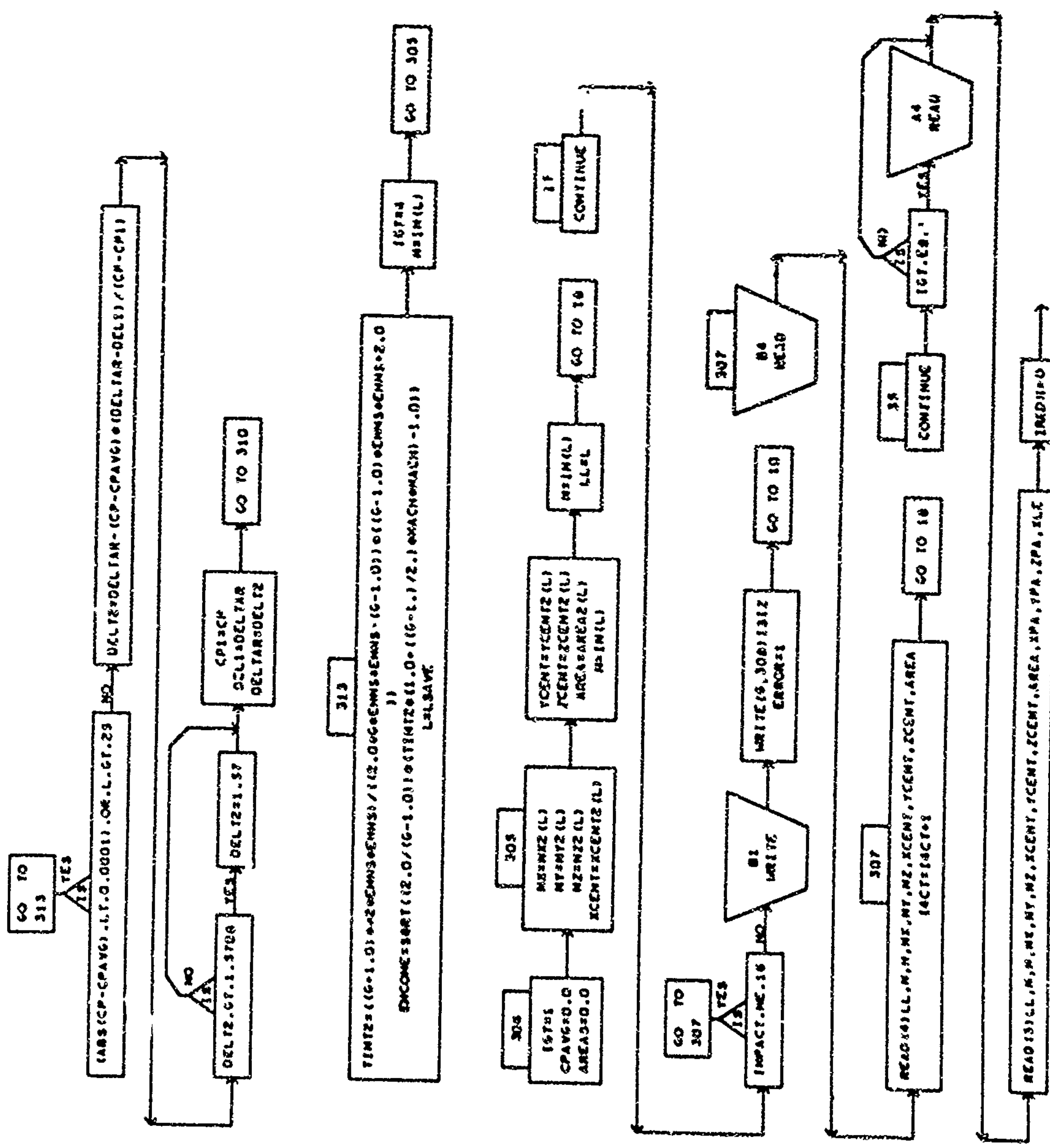
FORCE

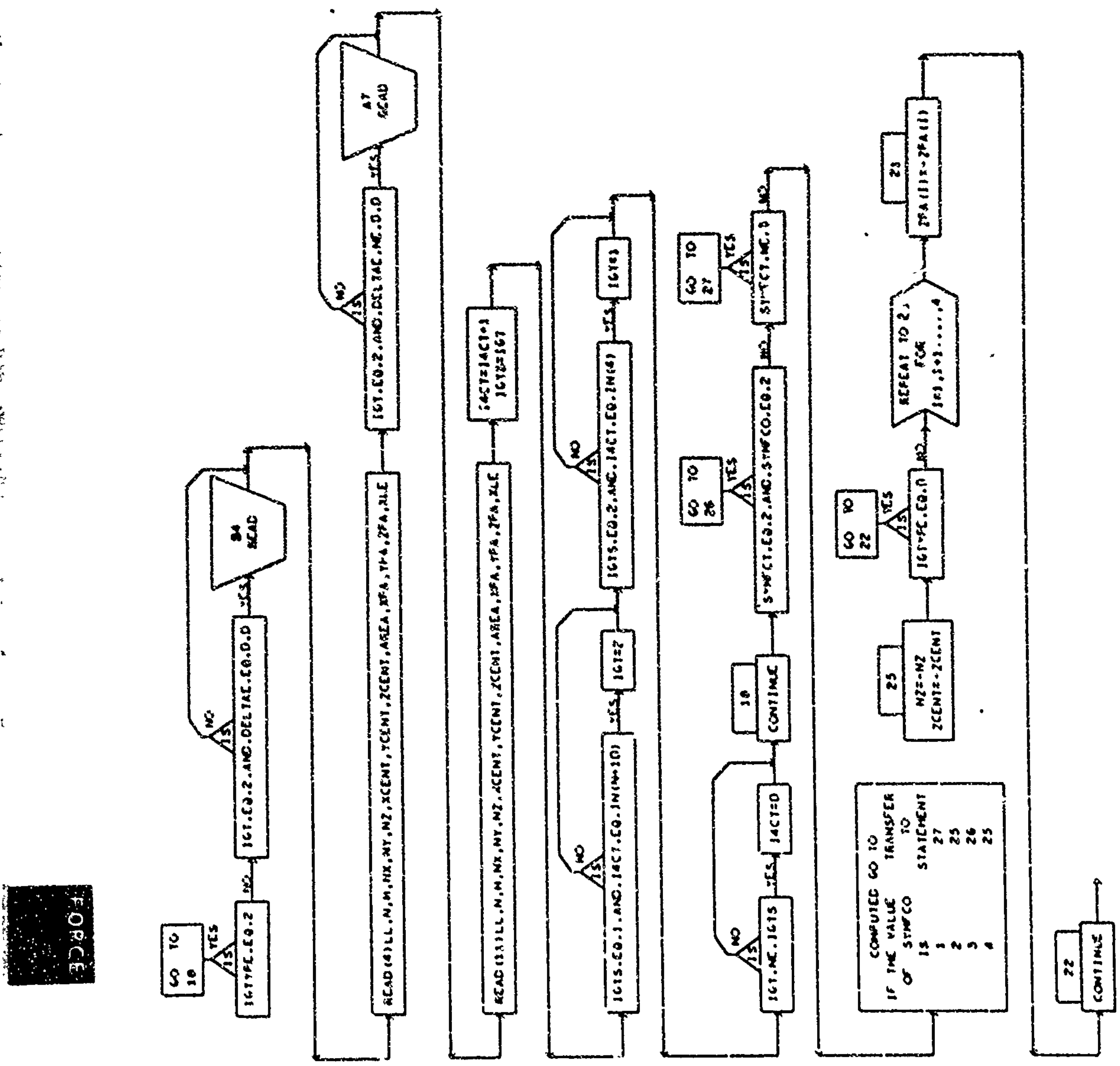
```
DECK AROI

      CCLM(J)   = CLM+ CLMI                                          AROI 6480
      CCLN(J)   = CLN+ CLNI                                          AROI 6490
C  RESOLVE NORMAL AND AXIAL FORCES IN LIFT AND DRAG DIRECTION        AROI 6500
      CD = CCA(J)*COS(ALPHAR)*COS(BETAR) - CCY(J)*SIN(BETAR)         AROI 6510
     1    +CCN(J)*SIN(ALPHAR)*COS(BETAR)                             AROI 6520
      CYPRIM = CCA(J)*COS(ALPHAR)*SIN(BETAR) + CCY(J)*COS(BETAR)     AROI 6530
     1        +CCN(J)*SIN(ALPHAR)*SIN(BETAR)                         AROI 6540
      CL = -CCA(J)*SIN(ALPHAR) + CCN(J)*COS(ALPHAR)                  AROI 6550
C  CALCULATE LIFT-DRAG RATIO                                         AROI 6560
      IF (CD .EQ. 0.0) CD = 0.000001                                 AROI 6570
      LOVERD = CL/CD                                                 AROI 6580
      CCL(J)   = CL                                                  AROI 6590
      CCD(J)   = CD                                                  AROI 6600
      CLOD(J) = LOVERD                                               AROI 6610
      CPS(J)   = CPSTAG                                              AROI 6620
      QQINFS(J)=QQINF                                                AROI 6630
C                                                                    AROI 6640
C                                                                    AROI 6650
   10 RETURN                                                         AROI 6660
      END                                                            AROI 6670
```

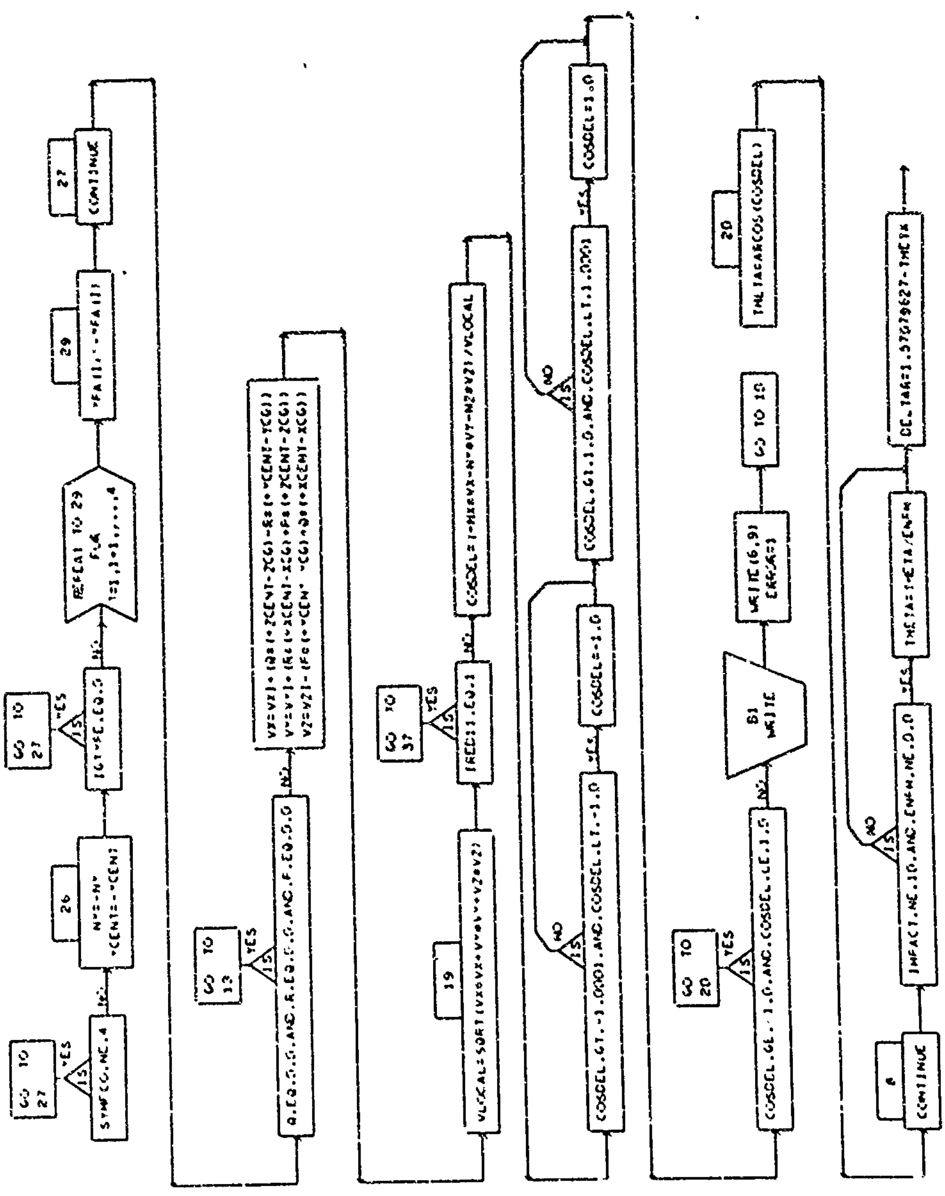
SUBROUTINE FORCE





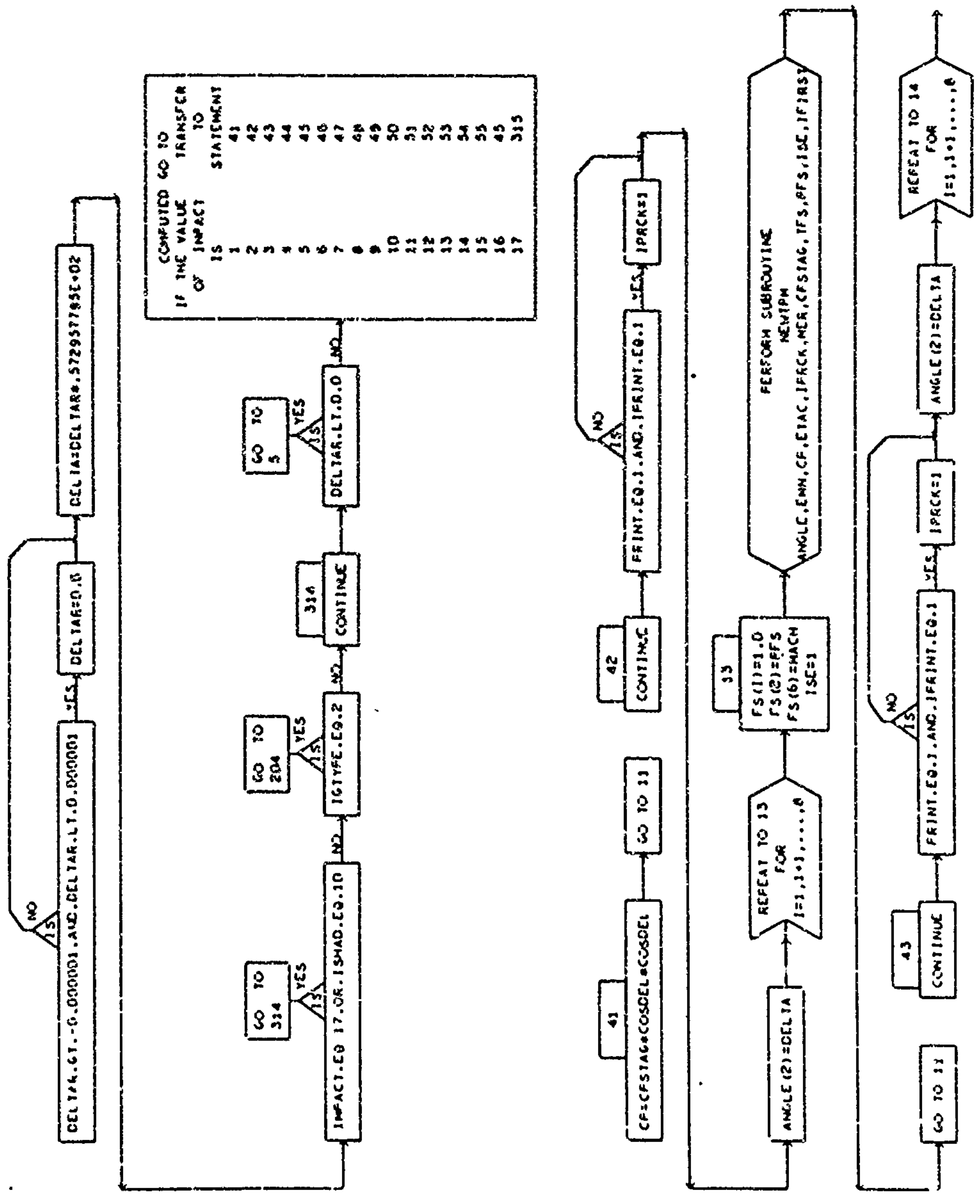


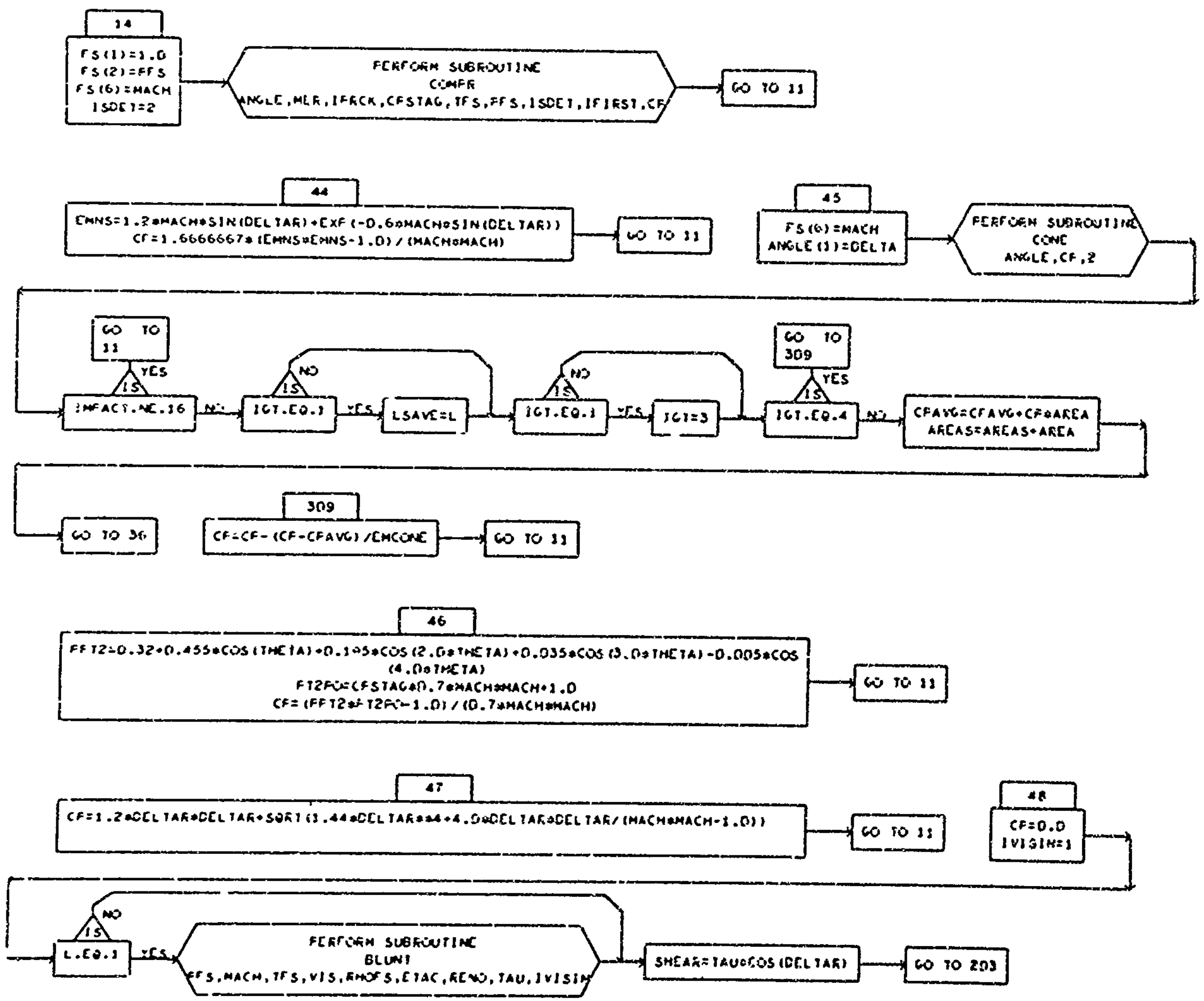

FORCE


GO TO 27
26
29
27
CONTINUE
REPEAT TO 29 FOR
GO TO 13
19
GO TO 37
COSDEL=1.0
COSDEL=-1.0
GO TO 20
81 WRITE
WRITE(6,91) ERRORS
GO TO 10
20
THETA=ARCOS(COSDEL)
CONTINUE
THETA=THETA/ENFM
IS
YES
NO

FORCE

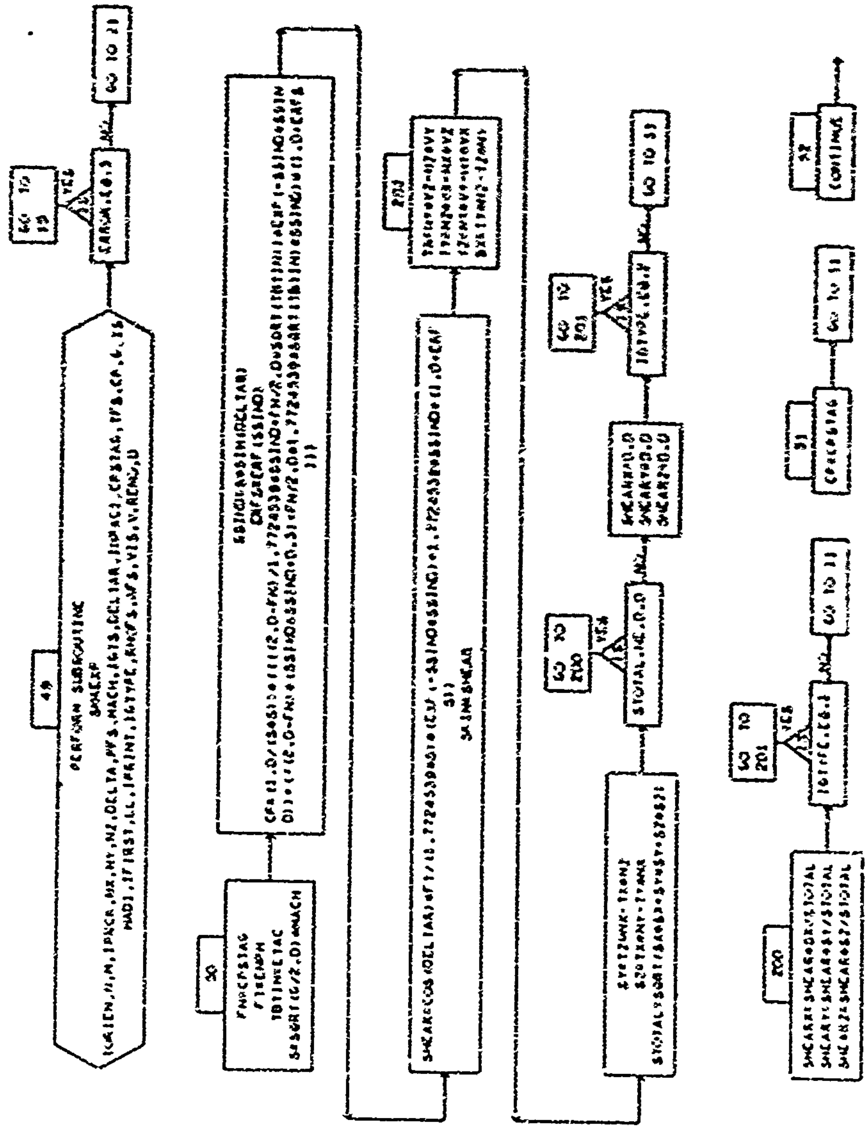
FORCE

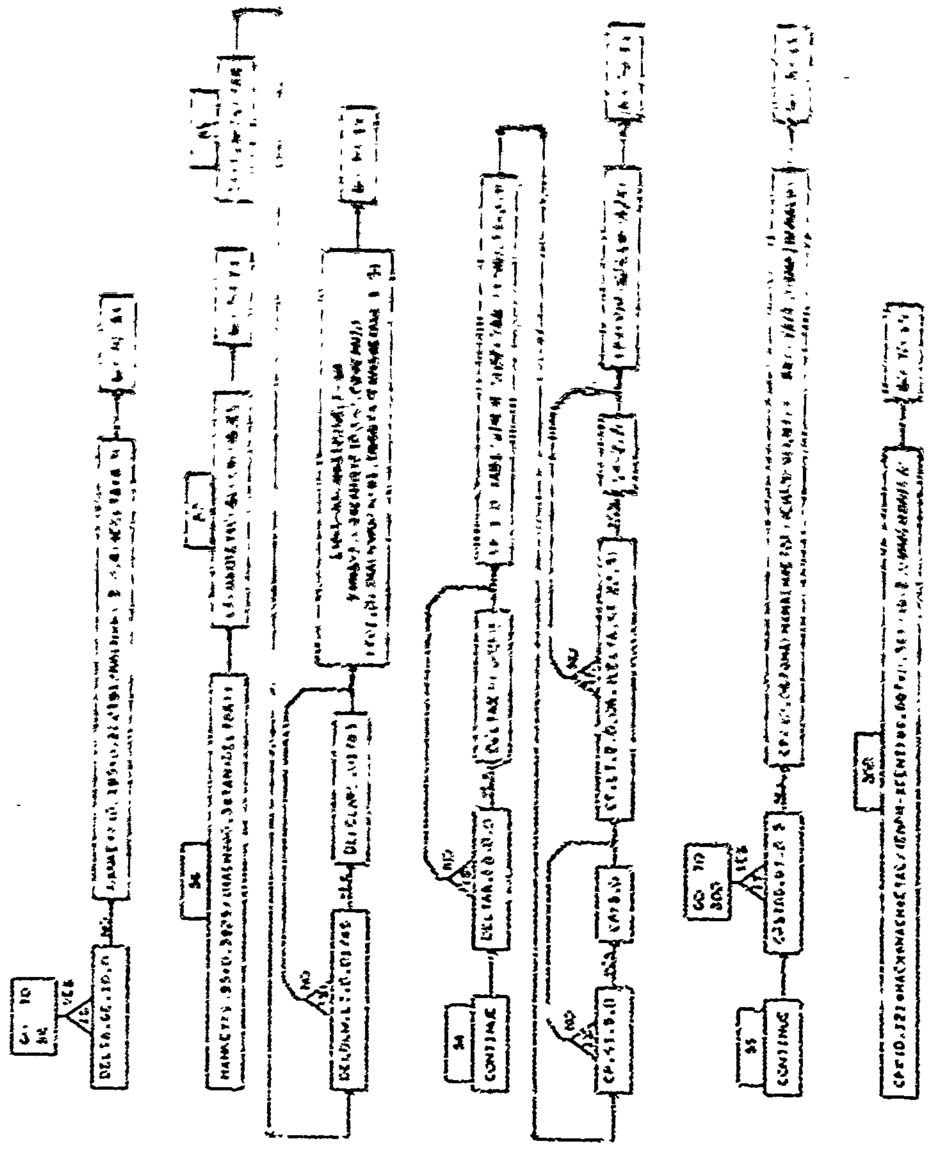

DELTA.GT.-0.000001.AND.DELTAR.LT.0.000001
IS
NO
YES
DELTAR=0.0
DELTA=DELTAR*.572957795E+02
IMPACT.EQ.17.OR.ISHAD.EQ.10
GO TO 314
IGTYPE.EQ.2
GO TO 204
314
CONTINUE
DELTAR.LT.0.0
GO TO 5
COMPUTED GO TO
IF THE VALUE OF IMPACT IS
TRANSFER TO STATEMENT
1 41
2 42
3 43
4 44
5 45
6 46
7 47
8 48
9 49
10 50
11 51
12 52
13 53
14 54
15 55
16 45
17 315
41
CF=CFSTAG*COSDEL*COSDEL
GO TO 11
42
CONTINUE
FRINT.EQ.1.AND.IFRINT.EQ.1
IPRCK=1
ANGLE(2)=DELTA
REPEAT TO 13 FOR I=1,1+1,....,6
13
FS(1)=1.0
FS(2)=EFS
FS(6)=MACH
ISE=1
PERFORM SUBROUTINE NEWTPH
ANGLE,EMN,CF,ETAC,IPRCK,MER,CFSTAG,TFS,EFS,ISE,IFIRST
GO TO 11
43
CONTINUE
FRINT.EQ.1.AND.IFRINT.EQ.1
IPRCK=1
ANGLE(2)=DELTA
REPEAT TO 14 FOR I=1,1+1,....,6

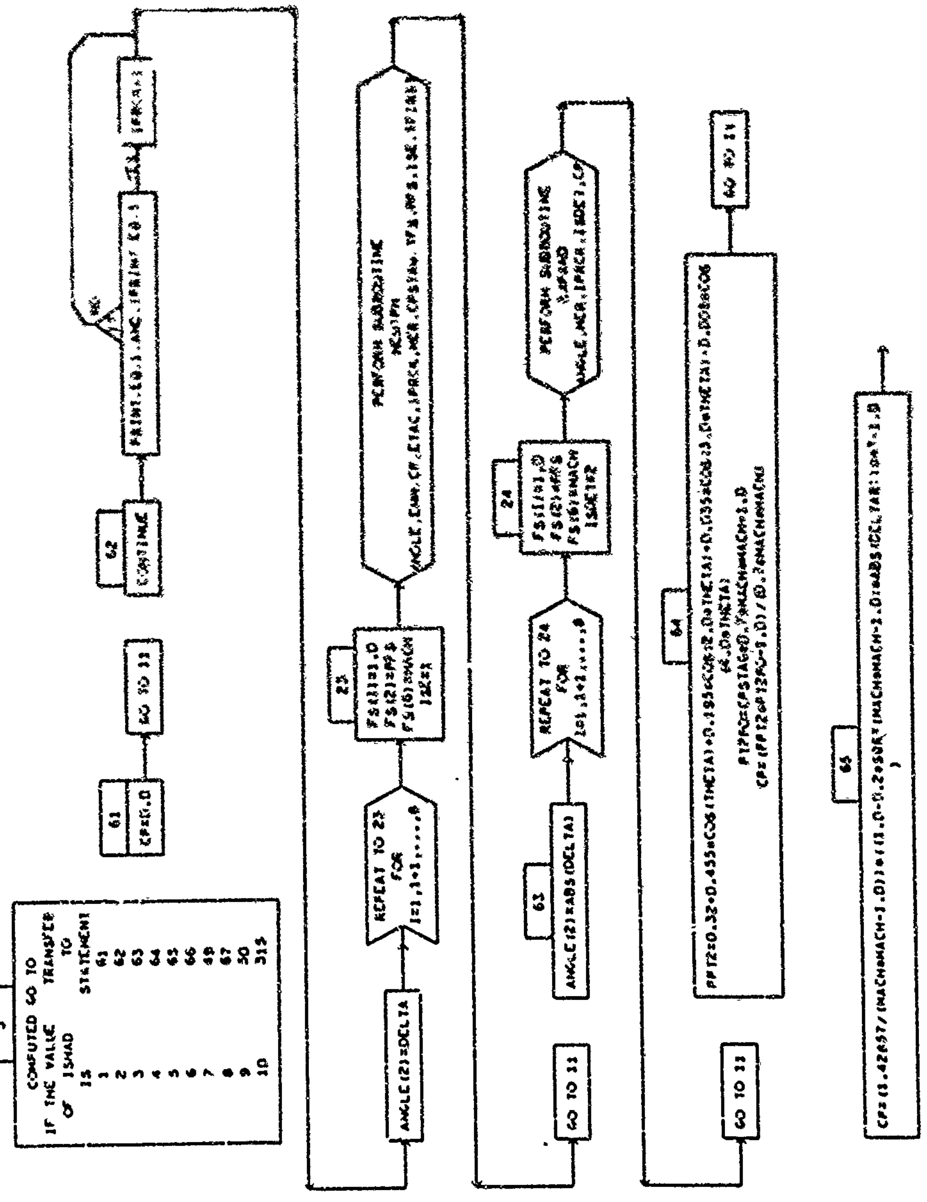


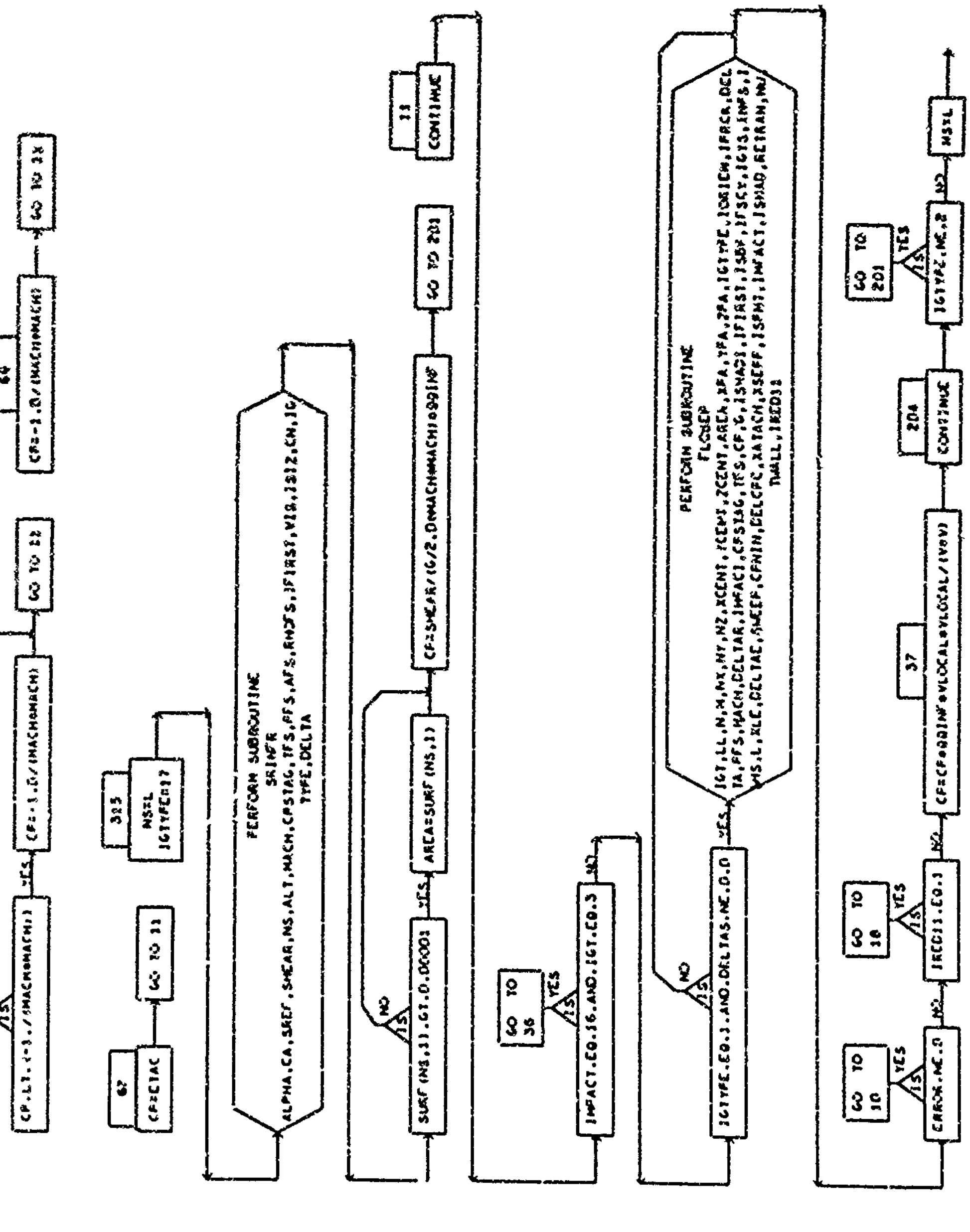
14
FS(1)=1.0
FS(2)=FFS
FS(6)=MACH
ISDET=2
PERFORM SUBROUTINE
COMPR
ANGLE,MER,IFRCK,CFSTAG,TFS,FFS,ISDET,IFIRST,CF
GO TO 11
44
EMNS=1.2*MACH*SIN(DELTAR)+EXP(-0.6*MACH*SIN(DELTAR))
CF=1.6666667*(EMNS*EMNS-1.0)/(MACH*MACH)
GO TO 11
45
FS(6)=MACH
ANGLE(1)=DELTA
PERFORM SUBROUTINE
CONE
ANGLE,CF,2
GO TO 11
YES
IS
IMPACT.NE.16
NO
IGT.EQ.1
YES
LSAVE=L
IGT.EQ.1
IGT=3
GO TO 309
IGT.EQ.4
CFAVG=CFAVG+CF*AREA
AREAS=AREAS+AREA
GO TO 36
309
CF=CF-(CF-CFAVG)/EMCONE
GO TO 11
46
FFT2=0.32+0.455*COS(THETA)+0.195*COS(2.0*THETA)+0.035*COS(3.0*THETA)-0.005*COS
(4.0*THETA)
FT2FO=CFSTAG*0.7*MACH*MACH+1.0
CF=(FFT2*FT2FO-1.0)/(0.7*MACH*MACH)
GO TO 11
47
CF=1.2*DELTAR*DELTAR+SQRT(1.44*DELTAR**4+4.0*DELTAR*DELTAR/(MACH*MACH-1.0))
GO TO 11
48
CF=0.0
IVISIN=1
L.EQ.1
PERFORM SUBROUTINE
BLUNT
FFS,MACH,TFS,VIS,RHOFS,ETAC,RENO,TAU,IVISIN
SHEAR=TAU*COS(DELTAR)
GO TO 203

FORCE

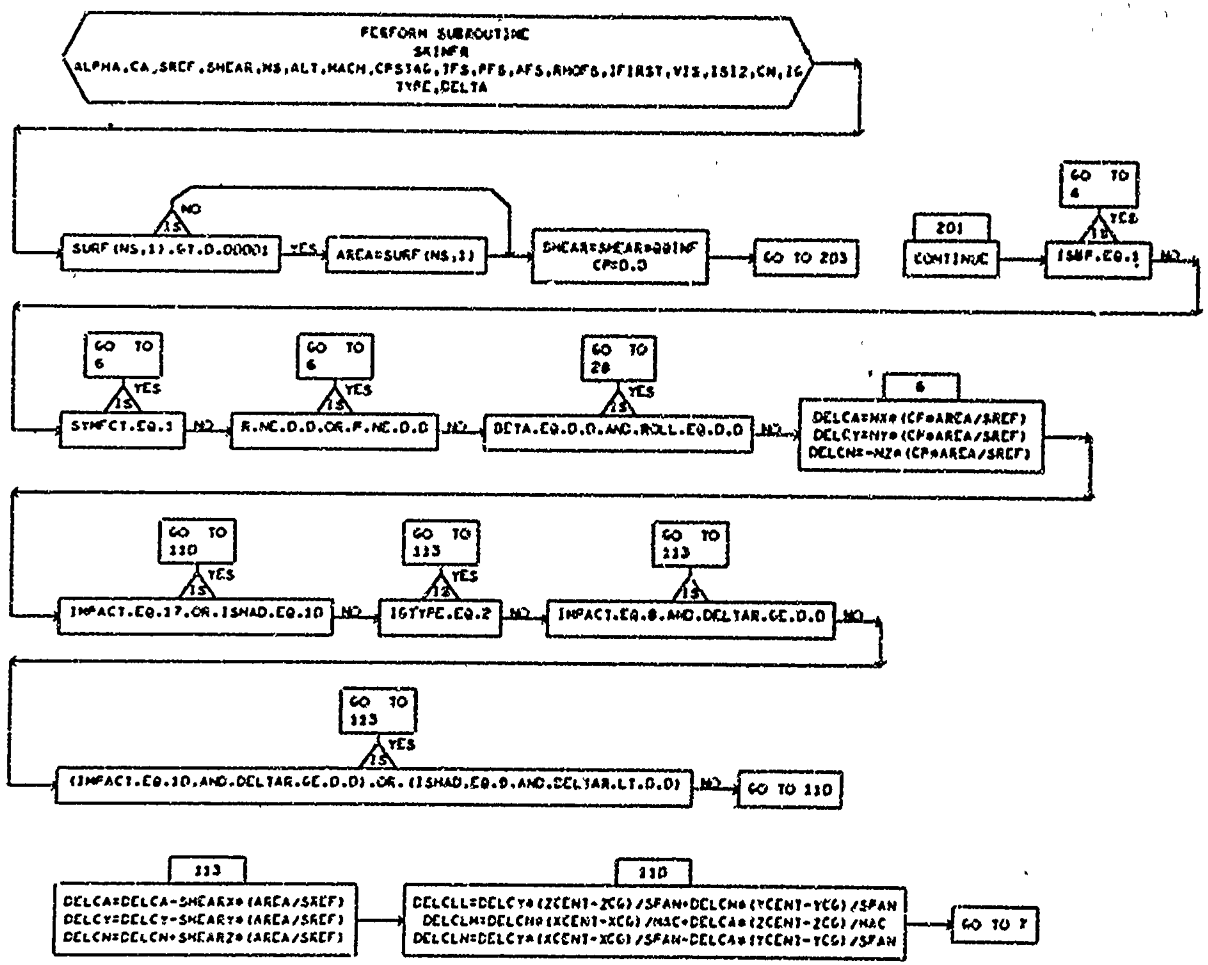


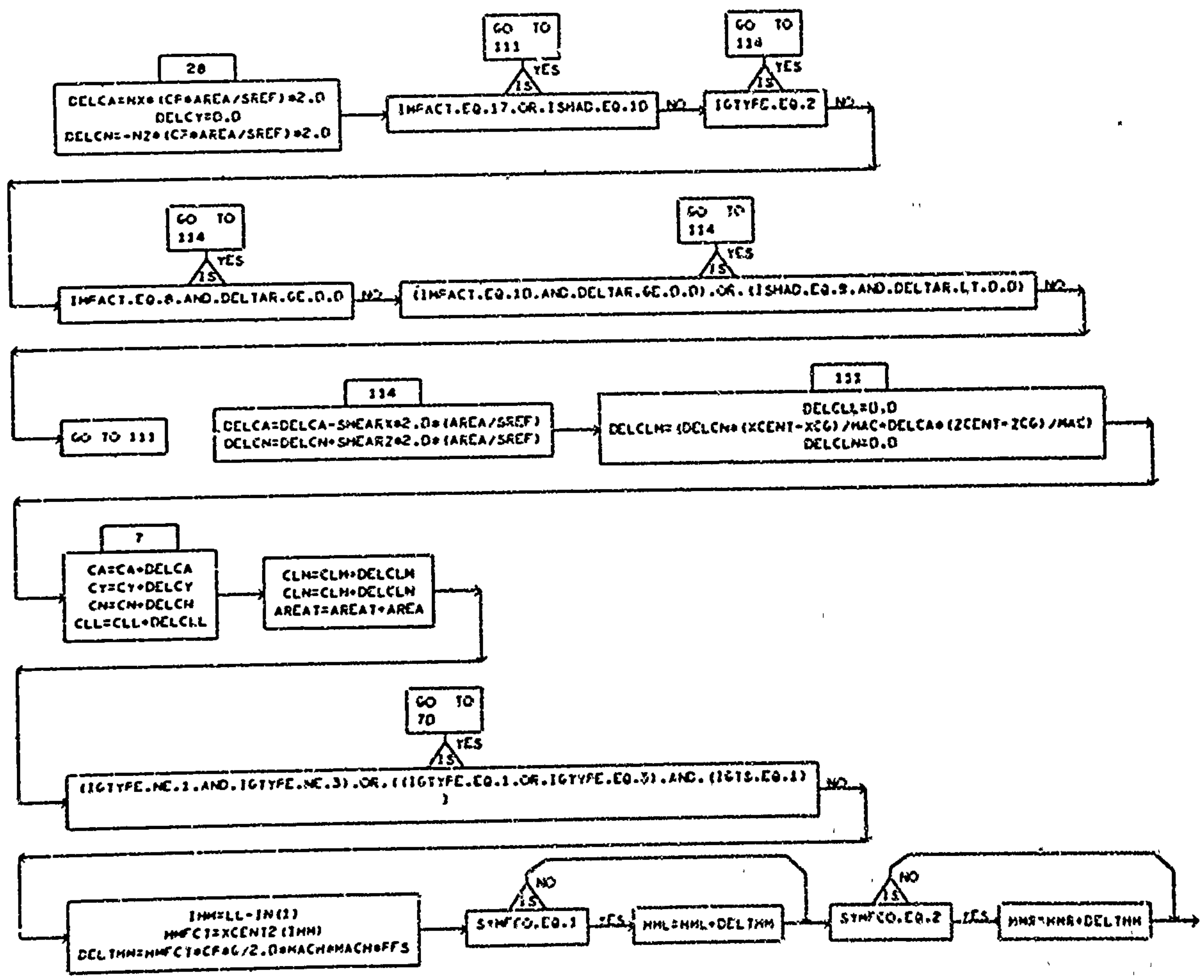



5
COMPUTED GO TO
IF THE VALUE OF ISHAD IS
TRANSFER TO STATEMENT
1 61
2 62
3 63
4 64
5 65
6 66
7 49
8 67
9 50
10 315
61
CP=0.0
GO TO 11
62
CONTINUE
NO
ANGLE(2)=DELTA
REPEAT TO 23 FOR
23
PERFORM SUBROUTINE
GO TO 11
63
ANGLE(2)=ABS(DELTA)
REPEAT TO 24 FOR
24
PERFORM SUBROUTINE
GO TO 11
64
GO TO 11
65

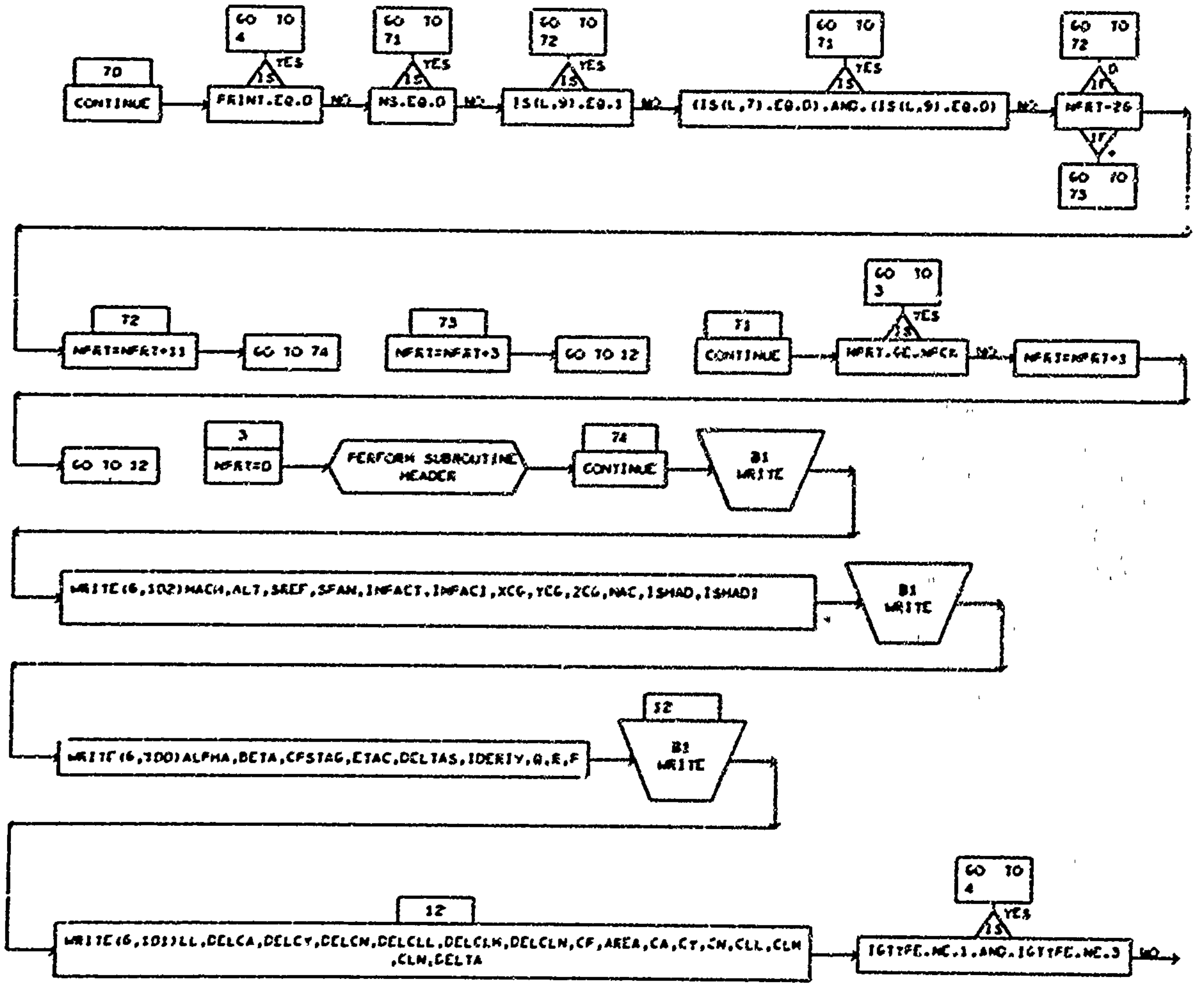

CP.LT.(-1./(MACH*MACH))
IS
NO
YES
CP=-1.D/(MACH*MACH)
GO TO 22
64
CP=-1.D/(MACH*MACH)
GO TO 2X
67
CP=CTAC
GO TO 11
325
NSTL
IGTYPE=17
PERFORM SUBROUTINE
SRINFR
ALPHA,CA,SREF,SHEAR,NS,ALT,MACH,CPSTAG,TFS,PFS,AFS,RHOFS,IFIRST,VIQ,ISI2,CN,IG
TYPE,DELTA
SURF(NS,1).GT.0.00001
AREA=SURF(NS,1)
CF=SHEAR/(G/2.D*MACH*MACH)*QQINF
GO TO 201
11
CONTINUE
GO TO
36
IMPACT.EQ.16.AND.IGT.EQ.3
IGTYPE.EQ.1.AND.DELTAS.NE.0.D
PERFORM SUBROUTINE
FLOSEP
IGT,LL,N,M,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,YPA,ZPA,IGTYPE,IORIEN,IFRCK,DEL
TA,PFS,MACH,DELTAR,IMPACI,CPSTAG,TFS,CF,G,ISHADI,IFIRST,ISBY,IFSCY,IGYS,INFS,I
NS,L,XLE,DELTAC,SWEEP,CPMIN,DELCPC,XATACH,XSEFF,ISPNT,IMPACT,ISHAD,RETRAN,NU
TWALL,IRED31
GO TO
10
ERROR.NE.0
GO TO
18
IRED31.EQ.1
37
CF=CF*QQINF*VLOCAL*VLOCAL/(V*V)
204
CONTINUE
GO TO
201
IGTYPZ.NE.2
NSTL

FORCE

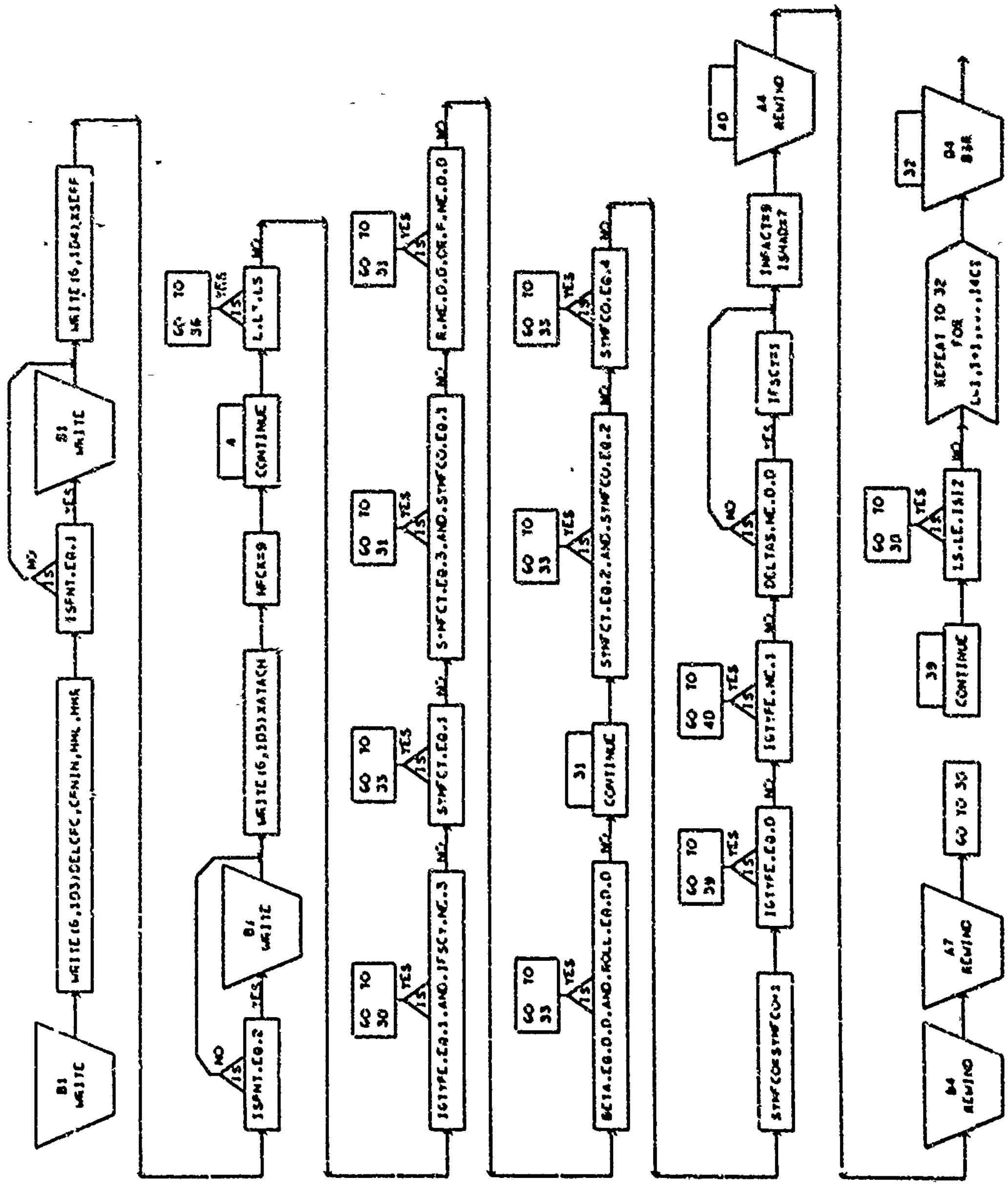

PERFORM SUBROUTINE
SKINFR
ALPHA,CA,SREF,SHEAR,NS,ALT,MACH,CPSTAG,IFS,PFS,AFS,RHOFS,IFIRST,VIS,IS12,CN,IG
TYPE,DELTA
SURF(NS,1).GT.0.00001
NO
YES
IS
AREA=SURF(NS,1)
SHEAR=SHEAR+QINF
CP=0.0
GO TO 203
201
CONTINUE
GO TO
4
IS
YES
ISMF.EQ.1
NO
GO TO
6
IS
YES
SYMFCT.EQ.1
NO
GO TO
6
IS
YES
R.NE.0.0.OR.F.NE.0.0
NO
GO TO
28
IS
YES
BETA.EQ.0.0.AND.ROLL.EQ.0.0
NO
6
DELCA=NX*(CP*AREA/SREF)
DELCY=NY*(CP*AREA/SREF)
DELCN=-NZ*(CP*AREA/SREF)
GO TO
110
IS
YES
IMPACT.EQ.17.OR.ISHAD.EQ.10
NO
GO TO
113
IS
YES
IGTYPE.EQ.2
NO
GO TO
113
IS
IMPACT.EQ.8.AND.DELTAR.GE.0.0
NO
GO TO
113
IS
YES
(IMPACT.EQ.10.AND.DELTAR.GE.0.0).OR.(ISHAD.EQ.9.AND.DELTAR.LT.0.0)
NO
GO TO 110
113
DELCA=DELCA-SHEARX*(AREA/SREF)
DELCY=DELCY-SHEARY*(AREA/SREF)
DELCN=DELCN+SHEARZ*(AREA/SREF)
110
DELCLL=DELCY*(ZCENT-ZCG)/SPAN+DELCN*(YCENT-YCG)/SPAN
DELCLM=DELCN*(XCENT-XCG)/MAC+DELCA*(ZCENT-ZCG)/MAC
DELCLN=DELCY*(XCENT-XCG)/SPAN-DELCA*(YCENT-YCG)/SPAN
GO TO 7

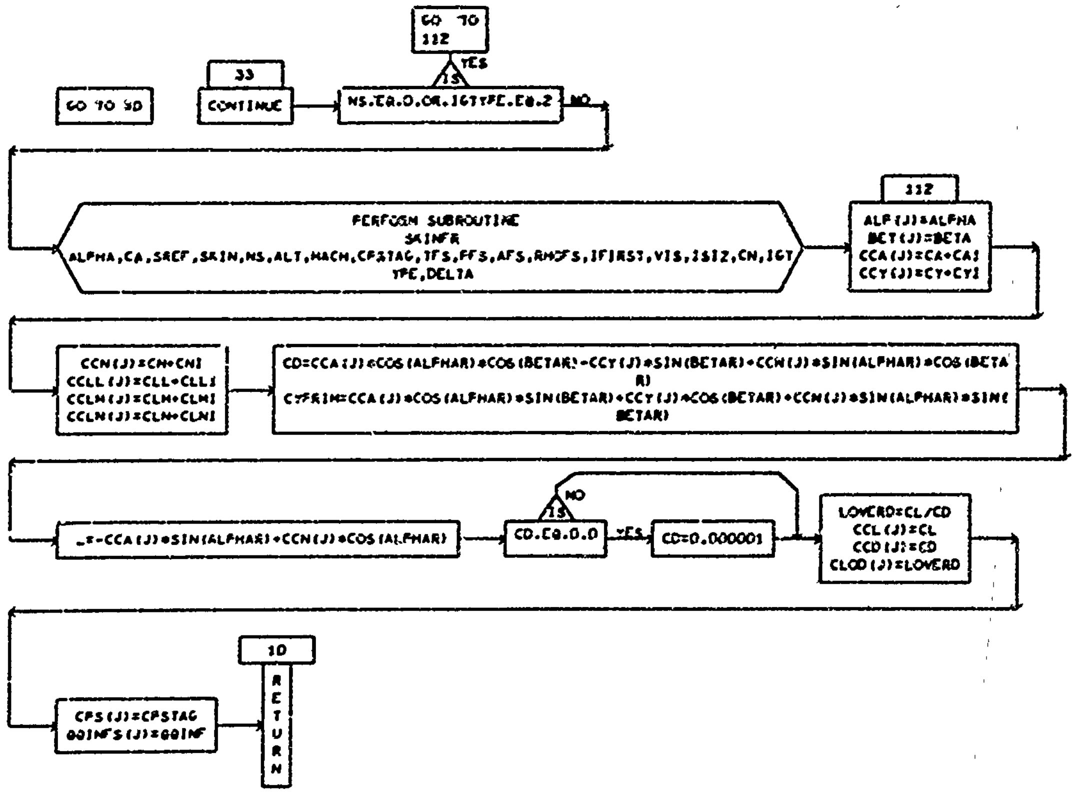

28
DELCA=NX*(CF*AREA/SREF)*2.0
DELCY=0.0
DELCN=-NZ*(CF*AREA/SREF)*2.0
IMPACT.EQ.17.OR.ISHAD.EQ.10
IS
YES
GO TO 111
NO
IGTYPE.EQ.2
IS
YES
GO TO 114
NO
IMPACT.EQ.8.AND.DELTAR.GE.0.0
IS
YES
GO TO 114
NO
(IMPACT.EQ.10.AND.DELTAR.GE.0.0).OR.(ISHAD.EQ.9.AND.DELTAR.LT.0.0)
IS
YES
GO TO 114
NO
GO TO 111
114
DELCA=DELCA-SHEARX*2.0*(AREA/SREF)
DELCN=DELCN+SHEARZ*2.0*(AREA/SREF)
111
DELCLL=0.0
DELCLM=(DELCN*(XCENT-XCG)/MAC+DELCA*(ZCENT-ZCG)/MAC)
DELCLN=0.0
7
CA=CA+DELCA
CY=CY+DELCY
CN=CN+DELCN
CLL=CLL+DELCLL
CLM=CLM+DELCLM
CLN=CLN+DELCLN
AREAT=AREAT+AREA
(IGTYPE.NE.1.AND.IGTYPE.NE.3).OR.((IGTYPE.EQ.1.OR.IGTYPE.EQ.3).AND.(IGTS.EQ.1)
)
IS
YES
GO TO 70
NO
IHM=LL-IN(1)
HMFCT=XCENT2(IHM)
DELTHM=HMFCT*CF*G/2.0*MACH*MACH*FFS
SYMFO.EQ.1
IS
NO
YES
HML=HML+DELTHM
SYMFO.EQ.2
IS
NO
YES
HMR=HMR+DELTHM

FORCE



70
CONTINUE
GO TO 4
YES
IS
PRINT.EQ.0
NO
GO TO 71
YES
IS
NS.EQ.0
NO
GO TO 72
YES
IS
IS(L,9).EQ.1
NO
GO TO 71
YES
IS
(IS(L,7).EQ.0).AND.(IS(L,9).EQ.0)
NO
GO TO 72
0
IF
NPRT-26
IF
+
GO TO 73
72
NPRT=NPRT+11
GO TO 74
73
NPRT=NPRT+3
GO TO 12
71
CONTINUE
GO TO 3
YES
IS
NPRT.GE.NPCR
NO
NPRT=NPRT+3
GO TO 12
3
NPRT=0
PERFORM SUBROUTINE HEADER
74
CONTINUE
B1 WRITE
WRITE(6,102)MACH,ALT,SREF,SFAN,INPACT,INPACI,XCG,YCG,ZCG,NAC,ISHAD,ISHADI
B1 WRITE
WRITE(6,100)ALPHA,BETA,CPSTAG,ETAC,DELTAS,IDERIV,Q,E,F
32
B1 WRITE
12
WRITE(6,101)LL,DELCA,DELCY,DELCN,DELCLL,DELCLM,DELCLN,CF,AREA,CA,CY,CN,CLL,CLM,CLN,DELTA
GO TO 4
YES
IS
IGTYPE.NE.1.AND.IGTYPE.NE.3
NO





GO TO 50
33
CONTINUE
NS.EQ.0.OR.IGTYPE.EQ.2
IS
YES
GO TO 112
NO
PERFORM SUBROUTINE
SKINFR
ALPHA,CA,SREF,SKIN,NS,ALT,MACH,CPSTAG,TFS,PFS,AFS,RHOFS,IFIRST,VIS,ISIZ,CN,IGTYPE,DELTA
112
ALF(J)=ALPHA
BET(J)=BETA
CCA(J)=CA+CAI
CCY(J)=CY+CYI
CCN(J)=CN+CNI
CCLL(J)=CLL+CLLI
CCLM(J)=CLM+CLMI
CCLN(J)=CLN+CLNI
CD=CCA(J)*COS(ALPHAR)*COS(BETAR)-CCY(J)*SIN(BETAR)+CCN(J)*SIN(ALPHAR)*COS(BETAR)
CYFRIN=CCA(J)*COS(ALPHAR)*SIN(BETAR)+CCY(J)*COS(BETAR)+CCN(J)*SIN(ALPHAR)*SIN(BETAR)
CL=-CCA(J)*SIN(ALPHAR)+CCN(J)*COS(ALPHAR)
CD.EQ.0.0
IS
NO
YES
CD=0.000001
LOVERD=CL/CD
CCL(J)=CL
CCD(J)=CD
CLOD(J)=LOVERD
CPS(J)=CPSTAG
QQINFS(J)=QQINF
10
RETURN

SYMBOLS USED IN SUBROUTINE FORCE

AFS	R	U	FREE-STREAM SPEED OF SOUND	FORCE
ALP	R	C	ANGLE OF ATTACK ARRAY	FORCE
ALPHA	R	A	ANGLE OF ATTACK, DEGREES	FORCE
ALPHAR	R	U	ANGLE OF ATTACK, RADIANS	FORCE
ALT	R	A	ALTITUDE, FEET	FORCE
ANGLE	R	D	FLOW ANGLE ARRAY	FORCE
AREA	R	U	ELEMENT AREA	FORCE
AREAS	R	U	TOTAL AREA OF A COLUMN OF ELEMENTS	FORCE
AREAT	R	U	TOTAL AREA	FORCE
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	FORCE
BET	R	C	YAW ANGLE ARRAY	FORCE
BETA	R	A	YAW ANGLE, DEGREES	FORCE
BETAR	R	U	YAW ANGLE, RADIANS	FORCE
BS	R	C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	FORCE
CA	R	U	AXIAL FORCE COEFFICIENT	FORCE
CAI	R	A	AXIAL FORCE INCREMENT	FORCE
CASE	I	C	CASE NUMBER	FORCE
CCA	R	C	AXIAL FORCE COEFFICIENT ARRAY	FORCE
CCD	R	C	DRAG COEFFICIENT ARRAY	FORCE
CCL	R	C	LIFT COEFFICIENT ARRAY	FORCE
CCLL	R	C	ROLLING MOMENT COEFFICIENT ARRAY	FORCE
CCLM	R	C	PITCHING MOMENT COEFFICIENT ARRAY	FORCE
CCLN	R	C	YAWING MOMENT COEFFICIENT ARRAY	FORCE
CCN	R	C	NORMAL FORCE COEFFICIENT ARRAY	FORCE
CCY	R	C	SIDE FORCE COEFFICIENT ARRAY	FORCE
CD	R	U	DRAG COEFFICIENT	FORCE
CF	R	C	SKIN FRICTION CONTRIBUTION TO AXIAL FORCE ARRAY	FORCE
CL	R	U	LIFT COEFFICIENT ARRAY	FORCE
CLL	R	U	ROLLING MOMENT COEFFICIENT	FORCE
CLLI	R	A	ROLLING MOMENT COEFFICIENT INCREMENT	FORCE
CLM	R	U	PITCHING MOMENT COEFFICIENT	FORCE
CLMI	R	A	PITCHING MOMENT COEFFICIENT INCREMENT	FORCE
CLN	R	U	YAWING MOMENT COEFFICIENT	FORCE
CLNI	R	A	YAWING MOMENT COEFFICIENT MOMENT	FORCE
CLOD	R	C	LIFT TO DRAG RATIO ARRAY	FORCE
CN	R	U	NORMAL FORCE COEFFICIENT	FORCE

FORCE

SYMBOLS USED IN SUBROUTINE FORCE

CNI	R	A	NORMAL FORCE COEFFICIENT INCREMENT	FORCE
COSDEL	R	U	COSINE OF ANGLE NORMAL AND VELOCITY VECTORS	FORCE
CP	R	U	PRESSURE COEFFICIENT	FORCE
CPAVG	R	U	AVERAGE PRESSURE COEFFICIENT TIMES AREA FOR EQUIVALENT CONE	FORCE
CPMIN	R	U	MINIMUN PRESSURE COEFFICIENT	FORCE
CPS	R	C	ARRAY FOR NEWTONIAN CORRELATION FACTOR, K	FORCE
CPSTAG	R	A	MODIFIED NEWTONIAN CORRELATION FACTOR, K	FORCE
CP1	R	U	ITERATION PRESSURE COEFFICIENT	FORCE
CY	R	U	SIDE FORCE COEFFICIENT	FORCE
CYI	R	A	SIDE FORCE COEFFICIENT INCREMENT	FORCE
CYPRIM	R	U	SIDE FORCE COEFFICIENT-WIND AXIS	FORCE
DELCA	R	U	ELEMENT CONTRIBUTION TO AXIAL FORCE	FORCE
DELCLL	R	U	ELEMENT CONTRIBUTION TO ROLLING MOMENT	FORCE
DELCLM	R	U	ELEMENT CONTRIBUTION TO PITCHING MOMENT	FORCE
DELCLN	R	U	ELEMENT CONTRIBUTION TO YAWING MOMENT	FORCE
DELCN	R	U	ELEMENT CONTRIBUTION TO NORMAL FORCE	FORCE
DELCPC	R	U	DELTA CP DUE TO CONTROL SURFACE DEFLECTION	FORCE
DELCY	R	U	ELEMENT CONTRIBUTION TO SIDE FORCE	FORCE
DELDLW	R	U	IMPACT ANGLE FOR DELTA WING	FORCE
DELTA	R	U	IMPACT ANGLE, DEGREES	FORCE
DELTAE	R	A	CONTROL SURFACE DEFLECTION	FORCE
DELTAR	R	U	IMPACT ANGLE, RADIANS	FORCE
DELTAS	R	U	IMPACT CONTROL SURFACE DEFLECTION ANGLE	FORCE
DELTHM	R	U	HINGE MOMENT INCREMENT	FORCE
DELT2	R	U	ITERATION VALUE FOR EQUIVALENT CONE ANGLE (2)	FORCE
DEL1	R	U	ITERATION VALUE FOR EQUIVALENT CONE ANGLE (1)	FORCE
EMCONE	R	U	MACH NUMBER ON SURFACE OF EQUIVALENT CONE	FORCE
EMN	R	U	MACH NUMBER TIMES SHOCK ANGLE SQUARED	FORCE
EMNS	R	U	MACH NUMBER NORMAL TO SHOCK	FORCE
ENPM	R	A	SURFACE SLOPE MODIFICATION FACTOR	FORCE
ERFS	R	U	ERROR FUNCTION PARAMETER	FORCE
ERROR	I	C	ERROR FLAG	FORCE
ETAC	R	A	PRANDTL-MEYER EXPANSION CORRECTION FACTOR	FORCE
FN	R	U	NORMAL MOMENTUM ACCOMODATION COEFFICIENT	FORCE
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	FORCE
FT	R	U	TANGENTIAL MOMENTUM ACCOMODATION COEFFICIENT	FORCE

SYMBOLS USED IN SUBROUTINE FORCE

G	R	U	RATIO OF SPECIFIC HEATS = 1.4	FORCE
HANKEY	R	U	NEWTONIAN CORRELATION FACTOR IN HANKEY EQUATION	FORCE
HMFCT	R	U	HINGE MOMENT FACTOR	FORCE
HML	R	A	HINGE MOMENT (+Y)	FORCE
HMR	R	A	HINGE MOMENT (-Y)	FORCE
HSIMP	R	U	HYPERSONIC INTERACTION PARAMETER	FORCE
IDERIV	I	A	DERIVATIVE CONTROL FLAG	FORCE
IFIRST	I	A	INITIAL PRINT FLAG FOR NEWTPM	FORCE
IFSCY	I	U	CONTROL SURFACE FLOW SEPARATION CALCULATION CYCLE NUMBER	FORCE
IGT	I	U	CONTROL SURFACE FLAG (=1 FORE-SURFACE, =2 CONTROL SURFACE)	FORCE
IGTS	I	U	CONTROL SURFACE FLAG FOR PRESENT ELEMENT	FORCE
IGTYPE	I	A	GEOMETRY TYPE (=1 FOR CONTROL SURFACE COMPOMENT)	FORCE
IHM	I	U	HINGE MOMENT ELEMENT INDEX	FORCE
IM	I	C	ELEMENT ROW NUMBER ARRAY	FORCE
IMPACI	I	A	STARTING ELEMENT IMPACT METHOD	FORCE
IMPACT	I	A	IMPACT FORCE CALCULATION METHOD	FORCE
IMPS	I	U	SAVED VALUE OF IMPACT FORCE METHOD	FORCE
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	FORCE
IORIEN	I	A	ELEMENT ORIENTATION	FORCE
IPRCK	I	U	PRINT FLAG	FORCE
IPRINT	I	A	PRINT FLAG	FORCE
IREDI1	I	U	TAPE 11 READ FLAG INDICATOR	FORCE
IS	I	C	SKIN FRICTION FLAG DATA ARRAY	FORCE
ISBP	I	U	FORCE SUMMATION BY PASS FLAG (=1 TO BY PASS SUMMATION)	FORCE
ISDET	I	U	DATA GENERATION CONTROL FLAG	FORCE
ISE	I	U	DATA GENERATION CONTROL FLAG	FORCE
ISHAD	I	A	SHADOW FORCE CALCULATION METHOD	FORCE
ISHADI	I	A	STARTING ELEMENT METHOD IN SHADOW REGION	FORCE
ISHS	I	U	SAVED VALUE OF SHADOW FORCE METHOD	FORCE
ISIZ	I	A	NUMBER OF ELEMENTS TO BE STORED IN CORE	FORCE
ISPNT	I	U	SEPARATION AND ATTACHMENT PRINT FLAG	FORCE
IVISIN	I	U	VISCOUS-INTERACTION CONTROL FLAG	FORCE
I4CT	I	U	COUNTER FOR READING CONTROL SURFACE GEOMETRY DATA	FORCE
J	I	A	ALPHA-BETA COUNTER FLAG	FORCE
K	I	C	NUMBER OF ELEMENTS	FORCE
L	I	U	ELEMENT FORCE CALCULATION LOOP COUNTER	FORCE
LL	I	U	ELEMENT NUMBER	FORCE

FORCE

SYMBOLS USED IN SUBROUTINE FORCE

LOVERD	R	U	LIFT TO DRAG RATIO	FORCE
LS	I	C	NUMBER OF ELEMENTS	FORCE
LSAVE	I	U	SAVEL ELEMENT NUMBER (ALSO ITERATION COUNTER)	FORCE
M	I	U	ELEMENT ROW NUMBER	FORCE
MAC	R	A	REFERENCE LENGTH FOR MOMENT COEFFICIENTS	FORCE
MACH	R	A	MACH NUMBER	FORCE
MER	I	U	ERROR FLAG	FORCE
N	I	U	ELEMENT COLUMN NUMBER	FORCE
NPCK	I	U	HEADER PRINT CHECK FLAG	FORCE
NPRT	I	U	PRINT COUNTER	FORCE
NS	I	A	NUMBER OF SKIN FRICTION SURFACES	FORCE
NW	R	A	WALL TEMPERATURE CALCULATION FLAG FOR FLOSEP	FORCE
NX	R	U	ELEMENT DIRECTION COSINE-X	FORCE
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	FORCE
NY	R	U	ELEMENT DIRECTION COSINE-Y	FORCE
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	FORCE
NZ	R	U	ELEMENT DIRECTION COSINE-Z	FORCE
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	FORCE
P	R	U	ROLL RATE	FORCE
PAGE	I	C	PAGE NUMBER	FORCE
PFS	R	A	FREE-STREAM PRESSURE-LBS / SQUARE FOOT	FORCE
PHIR	R	U	ROLL ANGLE, RADIANS	FORCE
PPT2	R	U	OSU METHOD PRESSURE RATIO	FORCE
PRINT	I	A	PRINT FLAG	FORCE
PSTAG	R	A	WIND TUNNEL STAGNATION PRESSURE-LBS / SQUARE FOOT	FORCE
PT2PO	R	U	OSU METHOD PRESSURE RATIO BEHIND NORMAL SHOCK	FORCE
Q	R	U	PITCH RATE, RADIANS / SECOND	FORCE
QQINF	R	A	DYNAMIC PRESSURE RATIO CORRECTION FACTOR	FORCE
QQINFS	R	C	SAVED VALUES OF DYNAMIC PRESSURE CORRECTION	FORCE
QRP	R	A	INPUT VEHICLE ROTATION RATE, RADIANS / SECOND	FORCE
R	R	U	YAW RATE, RADIANS / SECOND	FORCE
RENO	R	A	FREE STREAM REYNOLDS NUMBER	FORCE
RETRAN	R	A	TRANSITION REYNOLDS NUMBER FOR CONTROL SURFACE	FORCE
RHOFS	R	U	FREE STREAM DENSITY	FORCE
ROLL	R	U	ROLL ANGLE, DEGREES	FORCE
ROLLR	R	U	ROLL ANGLE, RADIANS	FORCE

SYMBOLS USED IN SUBROUTINE FORCE

S	R	U	FREE MOLECULAR FLOW SPEED RATIO	FORCE
SHEAR	R	U	FREE MOLECULAR FLOW SHEAR FORCE	FORCE
SHEARX	R	U	X-COMPONENT OF FREE MOLECULAR FLOW SHEAR FORCE	FORCE
SHEARY	R	U	Y-COMPONENT OF FREE MOLECULAR FLOW SHEAR FORCE	FORCE
SHEARZ	R	U	Z-COMPONENT OF FREE MOLECULAR FLOW SHEAR FORCE	FORCE
SKIN	R	U	TOTAL AXIAL FORCE SKIN FRICTION CONTRIBUTION	FORCE
SPAN	R	A	REFERENCE LENGTH FOR ROLLING, YAWING COEFFICIENTS	FORCE
SREF	R	A	VEHICLE REFERENCE AREA (WING AREA)	FORCE
SSIND	R	U	SPEED RATIO TIMES SINE OF IMPACT ANGLE	FORCE
STOTAL	R	U	TOTAL VALUE OF SHEAR FORCE VECTOR	FORCE
SURF	R	C	SKIN FRICTION DATA ARRAY	FORCE
SWEEP	R	A	LEADING EDGE SWEEP (NOT USED BY MARK II)	FORCE
SX	R	U	SHEAR FORCE VECTOR COMPONENT-X	FORCE
SY	R	U	SHEAR FORCE VECTOR COMPONENT-Y	FORCE
SYMFCO	I	U	SYMMETRY FACTOR	FORCE
SYMFCT	I	A	SYMMETRY FACTOR	FORCE
SZ	R	U	SHEAR FORCE VECTOR COMPONENT-Z	FORCE
TBTIN	R	U	RATIO OF BODY TEMPERATURE TO FREE-STREAM TEMPERATURE	FORCE
TFS	R	U	FREE-STREAM TEMPERATURE	FORCE
THETA	R	U	SURFACE SLOPE	FORCE
TINT2	R	U	TEMPERATURE PARAMETER FOR CONE MACH NUMBER EQUATION	FORCE
TITLE	R	C	TITLE	FORCE
TSTAG	R	A	WIND TUNNEL STAGNATION TEMPERATURE, DEGREES F	FORCE
TWALL	R	A	WALL TEMPERATURE FOR FLOSEP	FORCE
TX	R	U	FREE MOLECULAR FLOW VECTOR COMPONENT-X	FORCE
TY	R	U	FREE MOLECULAR FLOW VECTOR COMPONENT-Y	FORCE
TZ	R	U	FREE MOLECULAR FLOW VECTOR COMPONENT-Z	FORCE
V	R	A	FREE-STREAM VELOCITY, FEET / SECOND	FORCE
VIS	R	U	FREE-STREAM VISCOSITY	FORCE
VLOCAL	R	U	TOTAL LOCAL VELOCITY	FORCE
VX	R	U	LOCAL VELOCITY COMPONENT-X	FORCE
VXI	R	U	FREE-STREAM VELOCITY COMPONENT-X	FORCE
VY	R	U	LOCAL VELOCITY COMPONENT-Y	FORCE
VYI	R	U	FREE-STREAM VELOCITY COMPONENT-Y	FORCE
VZ	R	U	LOCAL VELOCITY COMPONENT-Z	FORCE
VZI	R	U	FREE STREAM VELOCITY COMPONENT-Z	FORCE

FORCE

SYMBOLS USED IN SUBROUTINE FORCE

XATACH	R	U	X-COORDINATE AT FLOW ATTACHMENT POINT	FORCE
XCENT	R	U	QUADRILATERAL ELEMENT CENTROID-X	FORCE
XCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-X	FORCE
XCG	R	A	X-CENTER FOR MOMENT CALCULATIONS	FORCE
XLE	R	U	X-DISTANCE FROM CENTRAID OF ELEMENT TO LEADING EDGE LINE	FORCE
XPA	R	D	X-COORDINATES OF QUADRILATERAL ELEMENT	FORCE
XSEPP	R	U	X-COORDINATE AT FLOW SEPARATION POINT	FORCE
YCENT	R	U	QUADRILATERAL ELEMENT CENTROID-Y	FORCE
YCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Y	FORCE
YCG	R	A	Y-CENTER FOR MOMENT CALCULATIONS	FORCE
YPA	R	D	Y-COORDINATES OF QUADRILATERAL ELEMENT	FORCE
ZCENT	R	U	QUADRILATERAL ELEMENT CENTRAID-Z	FORCE
ZCENT2	R	C	QUADRILATERAL ELEMENT CENTRAID ARRAY-Z	FORCE
ZCG	R	A	Z-CENTER FOR MOMENT CALCULATIONS	FORCE
ZPA	R	D	Z-COORDINATES OF QUADRILATERAL ELEMENT	FORCE

11. SUBROUTINE SHKEXP (DECK AROJ)

This subroutine performs a shock expansion analysis along a streamwise strip of elements.

a. Algorithm

The element is first checked to see if it is the first element in a strip. If it is, the local properties on it are calculated by the appropriate method and saved for use on the next element. For the next element in the strip the turning angle is calculated and either the compression or the expansion routine is used to determine its local properties (pressure coefficient, Mach number, and temperature).

b. Input/Output

None

c. Error

An error condition occurs when the number of initial strip elements is greater than 100, and when the wrong initial element method has been input.

d. Subroutines Required

COMPR, EXPAND, CONE

e. Argument List

(IORIEN, N, M, IPRCK, NX, NY, NZ, DELTA, PFS, MACH, IGTS, DELTAR, IMPACI, CPSTAG, TFS, CP, G, ISHADI, IFIRST, LL, IPRINT, IGTYPE, RHOFS, AFS, VIS, V, RENO, ISMODE)

f. Length

4472 bytes

DECK AROJ

```
      SUBROUTINE SHKEXP (IORIEN,N,M,IPRCK,NX,NY,NZ,DELTA,PFS,MACH,IGTS,
     1  DELTAR,IMPACI,CPSTAG,TFS,CP,G,ISHADI,IFIRST,LL,IPRINT,IGTYPE,
     2  RHOFS,AFS,VIS,V,RENO,ISMODE)
C
C*********************************************************************
C*** THIS SUBROUTINE CALCULATES LOCAL CONDITIONS ON AN ELEMENT USING ***
C*** SHOCK EXPANSION THEORY.                                        ***
C*********************************************************************
C
      DIMENSION  TITLE(15),ANGLE(3),FS(8),BS(8),NXI( 30),NYI( 30),
     1  NZI( 30),MACHI( 30),PI( 30),TI( 30)
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),
     1  YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)
C
      COMMON  CASF,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1  AREA2,IN,IM,K,LS,FS,BS
C
      INTEGER  CASF,PAGE,ERROR
C
      REAL  MACH,NX,NY,NZ,NXI,NYI,NZI,MACHI,NU
      REAL  NX2,NY2,NZ2
C
C
C  CHECK IF THIS ELEMENT IS THE FIRST ELEMENT IN A COLUMN OF ELEMENTS
      IF (IGTYPE .EQ. 1) GO TO 50
      IF( IORIEN.EQ.0.AND.N.EQ.1 .OR. IORIEN.EQ.1.AND.M.EQ.1 ) GO TO 80
      GO TO 51
   50 IF (IGTS.EQ.1 .AND. M.EQ.1) GO TO 80
C
C*********************************************************************
C******  CALCULATIONS FOR ELEMENTS THAT ARE NOT INITIAL ROW ELEMENTS ***
C*********************************************************************
   51 IF (IORIEN .EQ. 0) II = M
      IF (IORIEN.EQ.0 .AND. M.GT.MMAX) GO TO 80
      IF (IORIEN .EQ. 1) II=N
```

DECK AROJ

```
      SINNU =SQRT((NY*NZI(II)-NZ*NYI(II))**2+(NZ*NXI(II)-NX*NZI(II))**2
     1       +(NX*NYI(II)-NY*NXI(II))**2)
      NU = ARSIN(SINNU) * 0.5729578E02
      NU = ABS(NU)
      IF (NX .LT. NXI(II)) NU = -NU
      ANGLE(2) = ABS(NU)
C
C  SET UP DATA FOR COMPR OR EXPAND
      DO 82 I=1,8
   82 FS(I) = BS(I)
       FS(2) = PI(II)
       FS(3) = TI(II)
       FS(6) = MACHI(II)
       ISDET = 0
      IF (NU.GT.-0.00001 .AND. NU.LT.0.00001) GO TO 61
      IF (NU .LT. 0.0) GO TO 81
      CALL COMPR (ANGLE,MER,IPRCK,CPSTAG,TFS,PFS,ISDET,IFIRST,CP)
      GO TO 60
   81 CALL EXPAND (ANGLE,MER,IPRCK,ISDET,CP)
      GO TO 60
C
   61 DO 62 I=1,8
   62 BS(I) = FS(I)
   60 GO TO 97
C
C*********************************************************************
C************  CALCULATION FOR INITIAL ELEMENT OF EACH ROW  ***********
C*********************************************************************
C
C  CHECK IF THERE ARE TOO MANY INITIAL ELEMENTS (MAXIMUM OF 100)
   80 IF ((IORIEN.EQ.0 .AND. M.GT. 30).OR.(IORIEN.EQ.1.AND.N.GT. 30))
     1   GO TO 83
C
      MMAX = M
      IF (IORIEN.EQ.0) II=M
      IF (IORIEN.EQ.1) II=N
```

SHKEXP

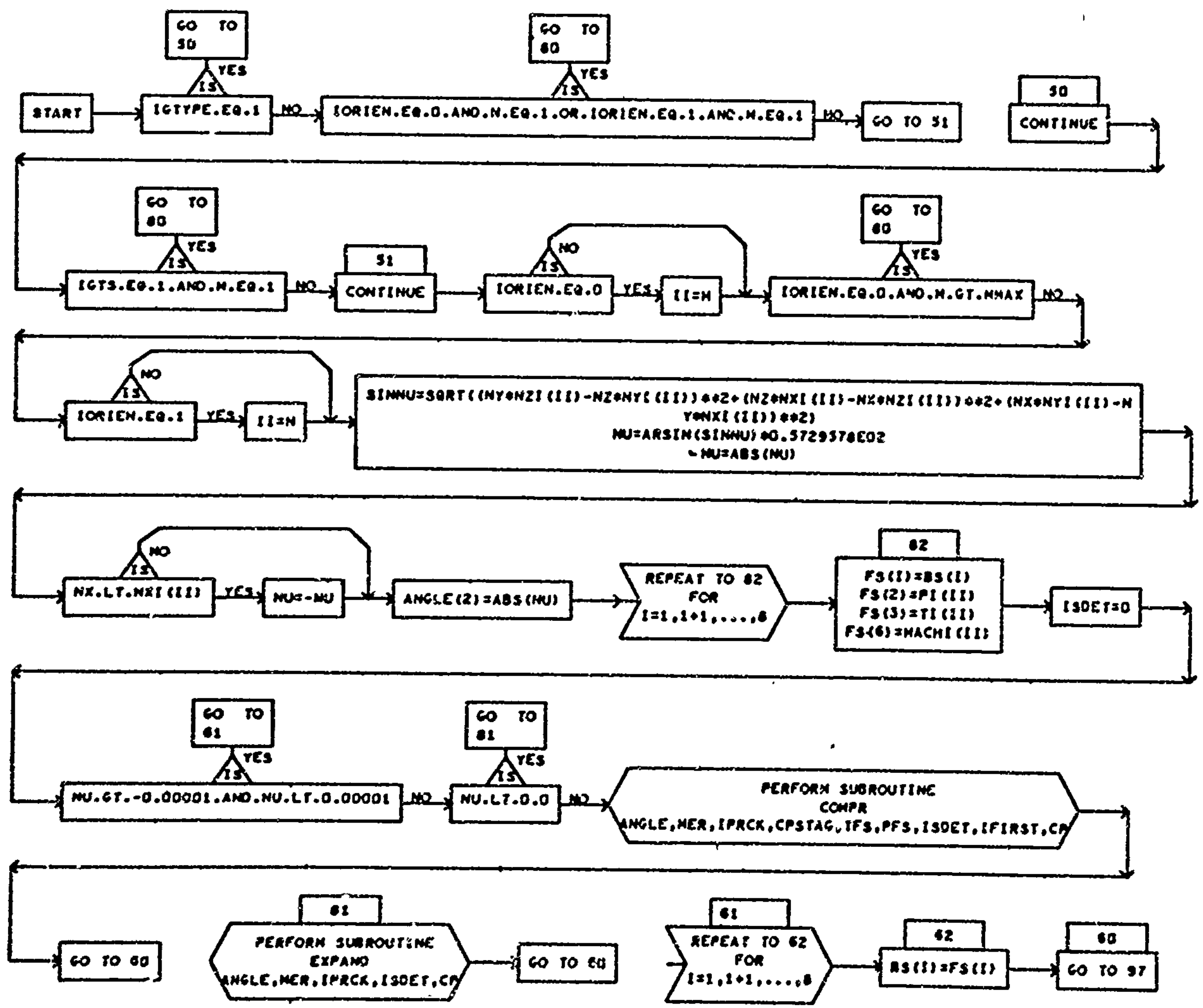
```
DECK AROJ

      IF (ISMODE .EQ. 1) GO TO 97
C
      ANGLE(2) = ABS(DELTA)
      NU = DELTA
      FS(1) = RHOFS
      FS(2) = PFS
      FS(3) = TFS
      FS(4) = AFS
      FS(5) = VIS
      FS(6) = MACH
      FS(7) = V
      FS(8) = RENO
      ISDET = 0
C
C  CHECK IF IMPACT OR SHADOW
      IF (DELTAR .LE. 0.0) GO TO 85
C
C  IMPACT FLOW  *****************
      GO TO (86,86,88,86,90,86,86,86,86,86,86,86,91),IMPACI
C
C  TANGENT WEDGE
   88 CALL COMPR (ANGLE,MER,IPRCK,CPSTAG,TFS,PFS,ISDET,IFIRST,CP)
      GO TO 97
C
C  TANGENT CONE EMPIRICAL
   90 ANGLE(1) = DELTA
      CALL CONE (ANGLE,CP,0)
      PI(II) = BS(2)
      MACHI(II) = BS(6)
      TI(II) = BS(3)
      GO TO 105
C  DELTA WING EMPIRICAL FOR SHOCK-EXPANSION CALCS.
   91 DELDLW = DELTAR
      IF (DELDLW .LT. 0.01745) DELDLW = 0.01745
      EMNS = MACH * SIN(DELDLW)
      EMNS = 1.09*EMNS + EXP(-0.49 *EMNS)
```

DECK AROJ

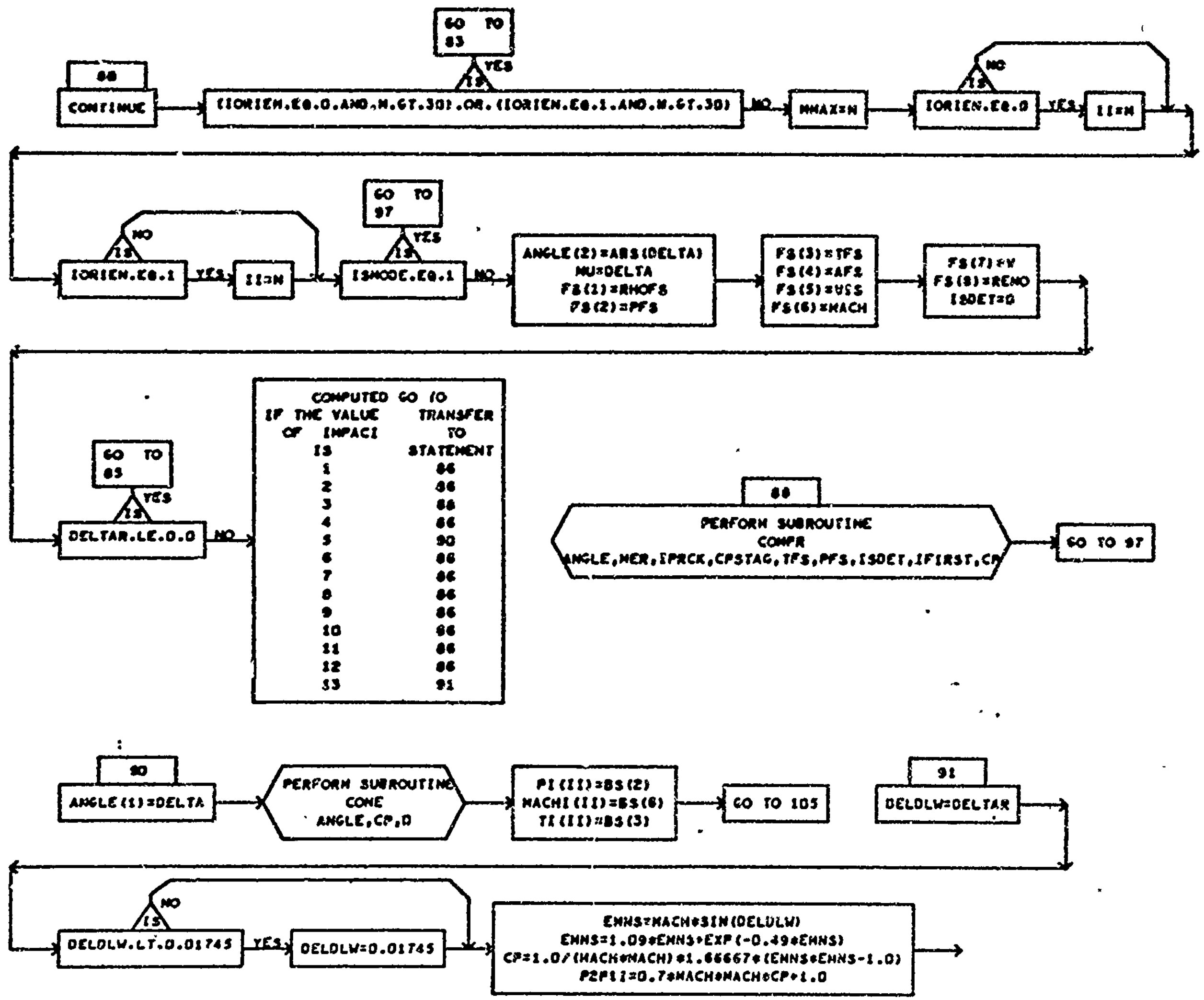
```
      CP = 1.0/(MACH*MACH) * 1.66667*(EMNS*EMNS-1.0)
      P2PII = 0.7*MACH*MACH*CP + 1.0
      PI(II) = P2PII * FS(2)
      TINT2 = ((G+1.0)**2 *EMNS*EMNS)/((2.0*G*EMNS*EMNS-(G-1.0))*
     1        ((G-1.0)*EMNS*EMNS + 2.0))
      MACHI(II) = SQRT((MACH*MACH-(4.0*(EMNS-1.0)*(G*EMNS+1.0))/
     1           ((G+1.0)**2*EMNS))*TINT2)
      TI(II) = FS(3) / TINT2
  105 BS(3) = TI(II)
      BS(2) = PI(II)
      BS(6) = MACHI(II)
      BS(1) = RHOFS * (BS(2)/PFS) * (TFS/BS(3))
      BS(4) = 49.021177 * SQRT(BS(3))
      IF (BS(3).GE.225.0) BS(5)=2.27*BS(3)**1.5/((BS(3)+198.6)*10.**8)
      IF (BS(3).LT.225.0) BS(5) = 0.80382436E-9 * BS(3)
      BS(7) = BS(4) * BS(6)
      BS(8) = BS(1) * BS(7) / BS(5)
      GO TO 99
C  SHADOW FLOW *************************
   85 GO TO (70,70,72,70,70,70,70),ISHADI
   70 GO TO 86
C  EXPANSION FROM FREESTREAM
   72 ISDET = 0
      CALL EXPAND (ANGLE,MER,IPRCK,ISDET,CP)
C
   97 MACHI(II) = BS(6)
      PI(II)    = BS(2)
      TI(II)    = BS(3)
      CP =  (BS(2)/PFS - 1.0) / (0.7*MACH*MACH)
   99 NXI(II) = NX
      NYI(II) = NY
      NZI(II) = NZ
C
C
   11 CONTINUE
      IF (IPRINT .EQ. 1) WRITE (6,100) LL,N,M,BS(2),BS(3),BS(6),CP,
```

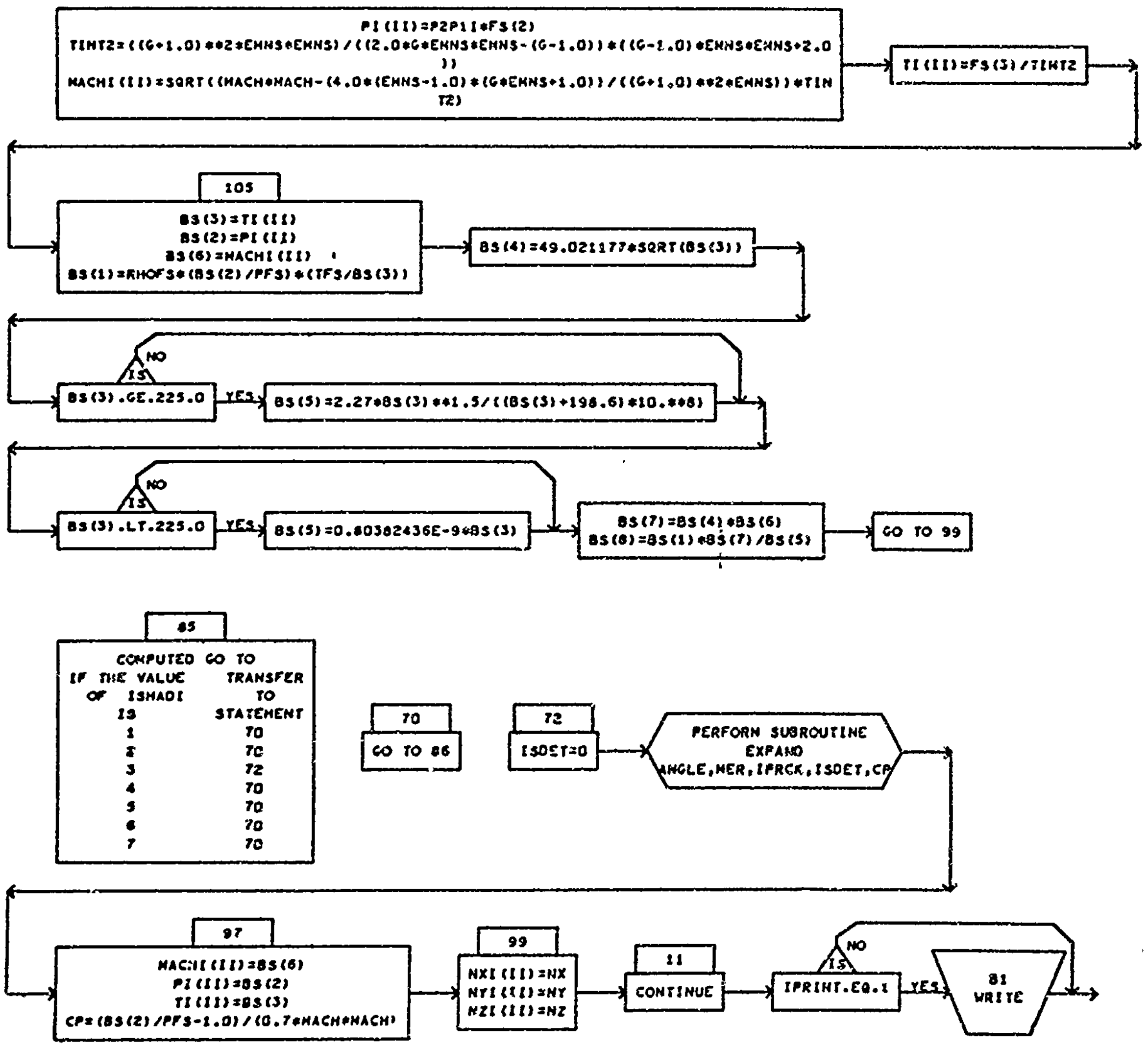
```
DECK AROJ

      1  NU,FS(2),FS(3),FS(6)
  100  FORMAT (1HO,32H  SHK-EXP. LOCAL CONDITIONS  LL=I5,4H  N=I4,
      1  4H  M=I4,4H  P=E12.5,4H  T=E12.5,7H  MACH=F7.3,5H  CP=E12.5,/
      2  1H ,28X,13HTURN ANGLE  =F9.4,4X,3HPI=E12.5,4H TI=E12.5,
      3  7H MACHI=F7.3 )
       RETURN
C
C
   83  WRITE (6,98)
   98  FORMAT (1H ,48H****NUMBER OF INITIAL ELEMENTS CANNOT EXCEED 100
      1 57H FOR SHOCK-EXPANSION CALCULATIONS.  CHANGE INPUT DATA**** )
       WRITE (6,101) IORIEN,IGTYPE,IGTS,LL,N,M
  101  FORMAT (1H ,10X,8HIORIEN =I2,5X,8HIGTYPE =I2,5X,6HIGTS =I2,
      1  5X,4HLL =I5,5X,3HN =I5,5X,3HM =I5)
       ERROR  = 3
       RETURN
   86  WRITE(6,96)
   96  FORMAT (1H ,47H****DURING SHOCK-EXPANSION CALCULATIONS PROGRAM
      1 59H TRIED TO USE WRONG INITIAL ELEMENT METHOD--CHECK INPUT****)
       ERROR  = 3
       RETURN
C
       END
```

SUBROUTINE SHKEXP

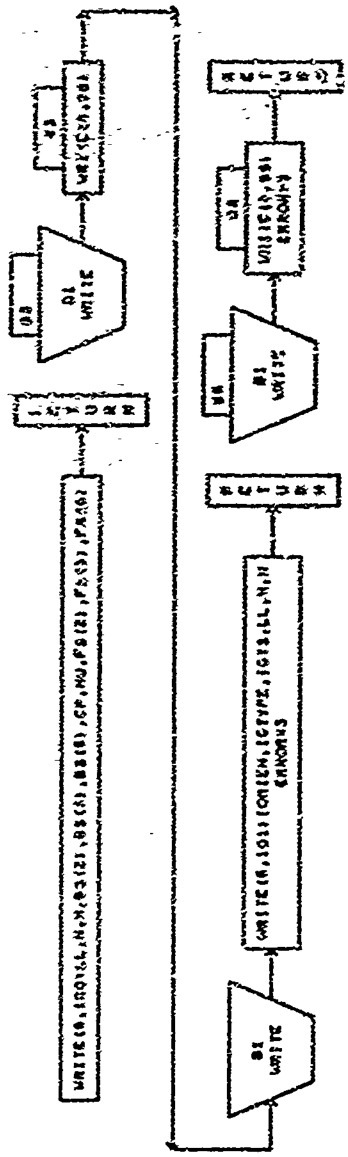


SHKEXP


CONTINUE
((IORIEN.EQ.0.AND.N.GT.30).OR.((IORIEN.EQ.1.AND.N.GT.30)
IS
YES
GO TO 83
NO
NMAX=N
IORIEN.EQ.0
IS
NO
YES
II=N
IORIEN.EQ.1
IS
NO
YES
II=N
ISMODE.EQ.1
IS
YES
GO TO 97
NO
ANGLE(2)=ABS(DELTA)
MU=DELTA
FS(1)=RHOFS
FS(2)=PFS
FS(3)=TFS
FS(4)=AFS
FS(5)=VSS
FS(6)=MACH
FS(7)=V
FS(8)=RENO
ISDET=0
DELTAR.LE.0.0
IS
YES
GO TO 85
NO
COMPUTED GO TO
IF THE VALUE OF IMPACI IS / TRANSFER TO STATEMENT
1 86
2 86
3 86
4 86
5 90
6 86
7 86
8 86
9 86
10 86
11 86
12 86
13 91
PERFORM SUBROUTINE
COMPR
ANGLE,MER,IPRCK,CPSTAG,TFS,PFS,ISDET,IFIRST,CP
GO TO 97
90
ANGLE(1)=DELTA
PERFORM SUBROUTINE
CONE
ANGLE,CP,D
PI(II)=BS(2)
MACHI(II)=BS(6)
TI(II)=BS(3)
GO TO 105
91
DELDLW=DELTAR
DELDLW.LT.0.01745
IS
NO
YES
DELDLW=0.01745
EMNS=MACH*SIN(DELDLW)
EMNS=1.09*EMNS+EXP(-0.49*EMNS)
CP=1.0/(MACH*MACH)*1.66667*(EMNS*EMNS-1.0)
P2P1=0.7*MACH*MACH*CP+1.0


PI(II)=P2P1I*FS(2)
TINT2=((G+1.0)**2*EMNS*EMNS)/((2.0*G*EMNS*EMNS-(G-1.0))*((G-1.0)*EMNS*EMNS+2.0))
MACHI(II)=SQRT((MACH*MACH-(4.0*(EMNS-1.0)*(G*EMNS+1.0))/((G+1.0)**2*EMNS))*TINT2)
TI(II)=FS(3)/TINT2
105
BS(3)=TI(II)
BS(2)=PI(II)
BS(6)=MACHI(II)
BS(1)=RHOFS*(BS(2)/PFS)*(TFS/BS(3))
BS(4)=49.021177*SQRT(BS(3))
IS
NO
BS(3).GE.225.0
YES
BS(5)=2.27*BS(3)**1.5/((BS(3)+198.6)*10.**8)
IS
NO
BS(3).LT.225.0
YES
BS(5)=0.80382436E-9*BS(3)
BS(7)=BS(4)*BS(6)
BS(8)=BS(1)*BS(7)/BS(5)
GO TO 99
85
COMPUTED GO TO
IF THE VALUE OF ISHADI IS / TRANSFER TO STATEMENT
1 70
2 70
3 72
4 70
5 70
6 70
7 70
70
GO TO 86
72
ISDET=0
PERFORM SUBROUTINE
EXPAND
ANGLE,MER,IFRCK,ISDET,CP
97
MACHI(II)=BS(6)
PI(II)=BS(2)
TI(II)=BS(3)
CP=(BS(2)/PFS-1.0)/(0.7*MACH*MACH)
99
NXI(II)=NX
NYI(II)=NY
NZI(II)=NZ
11
CONTINUE
IS
NO
IPRINT.EQ.1
YES
81
WRITE

SHKEXP


WRITE
RETURN
WRITE
WRITE
RETURN
WRITE
RETURN

SYMBOLS USED IN SUBROUTINE SHKEXP

AFS	R	A	FREE-STREAM SPEED OF SOUND	SHKEXP
ANGLE	R	D	FLOW ANGLE ARRAY	SHKEXP
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	SHKEXP
BS	R	C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	SHKEXP
CASE	I	C	CASE NUMBER	SHKEXP
CP	R	A	PRESSURE COEFFICIENT	SHKEXP
CPSTAG	R	A	MODIFIED NEWTONIAN CORRELATION FACTOR, K	SHKEXP
DELDLW	R	U	IMPACT ANGLE FOR DELTA-WING OPTION, RADIANS	SHKEXP
DELTA	R	A	IMPACT ANGLE, DEGREES	SHKEXP
DELTAR	R	A	IMPACT ANGLE, RADIANS	SHKEXP
EMNS	R	U	MACH NORMAL TO THE SHOCK	SHKEXP
ERROR	I	C	ERROR FLAG	SHKEXP
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	SHKEXP
G	R	A	RATIO OF SPECIFIC HEATS	SHKEXP
IFIRST	I	A	FIRST POINT FLAG FOR USE IN NEWTPM	SHKEXP
IGTS	I	A	CONTROL SURFACE FLAG FOR PRESENT ELEMENT	SHKEXP
IGTYPE	I	A	GEOMETRY TYPE (=1 FOR CONTROL SURFACE COMPONENT)	SHKEXP
II	I	U	LEADING ELEMENT INDEX	SHKEXP
IM	I	C	ELEMENT ROW NUMBER ARRAY	SHKEXP
IMPACI	I	A	STARTING ELEMENT IMPACT METHOD	SHKEXP
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	SHKEXP
IORIEN	I	A	ELEMENT ORIENTATION	SHKEXP
IPRCK	I	A	PRINT FLAG	SHKEXP
IPRINT	I	A	PRINT FLAG	SHKEXP
ISDET	I	U	DATA GENERATION CONTROL FLAG	SHKEXP
ISHADI	I	A	STARTING ELEMENT METHOD IN SHADOW REGION	SHKEXP
ISMODE	I	A	SHOCK-EXPANSION MODE FLAG (USED IN FLOSEP)	SHKEXP
K	I	C	NUMBER OF ELEMENTS	SHKEXP
LL	I	A	ELEMENT NUMBER	SHKEXP
LS	I	C	NUMBER OF ELEMENTS	SHKEXP
M	I	A	ELEMENT ROW NUMBER	SHKEXP
MACH	R	A	MACH NUMBER	SHKEXP
MACHI	R	D	STARTING OR PREVIOUS ELEMENT MACH NUMBER	SHKEXP
MER	I	U	ERROR FLAG	SHKEXP
MMAX	I	U	MAXIMUM VALUE FOR PARAMETER M	SHKEXP
N	I	A	ELEMENT COLUMN NUMBER	SHKEXP

SYMBOLS USED IN SUBROUTINE SHKEXP

```
NU       R  U  FLOW TURNING ANGLE                                 SHKEXP
NX       R  A  ELEMENT DIRECTION COSINE-X                         SHKEXP
NX1      R  D  ELEMENT DIRECTION COSINE-X                         SHKEXP
NX2      R  C  ELEMENT DIRECTION COSINE ARRAY-X                   SHKEXP
NY       R  A  ELEMENT DIRECTION COSINE-Y                         SHKEXP
NY1      R  D  ELEMENT DIRECTION COSINE-Y                         SHKEXP
NY2      R  C  ELEMENT DIRECTION COSINE ARRAY-Y                   SHKEXP
NZ       R  A  ELEMENT DIRECTION COSINE-Z                         SHKEXP
NZ1      R  D  ELEMENT DIRECTION COSINE-Z                         SHKEXP
NZ2      R  C  ELEMENT DIRECTION COSINE ARRAY-Z                   SHKEXP
PAGE     I  C  PAGE NUMBER                                        SHKEXP
PFS      R  A  FREE-STREAM PRESSURE, LBS/SQUARE FOOT              SHKEXP
P1       R  D  PRESSURE                                           SHKEXP
P2P11    R  U  PRESSURE RATIO                                     SHKEXP
RENO     R  A  FREE-STREAM REYNOLDS NUMBER                        SHKEXP
RHOFS    R  A  FREE-STREAM DENSITY                                SHKEXP
SINNU    R  U  SINE OF FLOW TURNING ANGLE                         SHKEXP
TFS      R  A  FREE STREAM TEMPERATURE                            SHKEXP
T1       R  D  LOCAL TEMPERATURE                                  SHKEXP
TINT2    R  U  TEMPERATURE RATIO                                  SHKEXP
TITLE    R  C  TITLE                                              SHKEXP
V        R  A  FREE STREAM VELOCITY, FEET/SECOND                  SHKEXP
VIS      R  A  FREE-STREAM VISCOSITY                              SHKEXP
XCENT2   R  C  QUADRILATERAL ELEMENT CENTROID ARRAY-X             SHKEXP
YCENT2   R  C  QUADRILATERAL ELEMENT CENTROID ARRAY-Y             SHKEXP
ZCENT2   R  C  QUADRILATERAL ELEMENT CENTROID ARRAY-Z             SHKEXP
```

12. SUBROUTINE FLOSEP (DECK AROK)

This subroutine has the task of determining the effect of flow separation caused by the deflection of a control surface.

a. Algorithm

The flow separation calculations are performed in three cycles. The first cycle is with the control deflected and determines if separation will occur. The second cycle determines where the separation occurs and the pressure changes due to separation. The third cycle calculated the final pressure, using the normal program input method in combination with the flow separation increments.

b. Input/Output

None

c. Error

An error condition occurs when the number of streamwise elements for the control fore-surface and flap is greater than 125.

d. Subroutines Required

TEMP, COMPR, SHKEXP

e. Argument List

(IGT, LL, N, M, NX, NY, NZ, XCENT, YCENT, ZCENT, AREA, XPA, YPA, ZPA, IGTYPE, IORIEN, IPRCK, DELTA, PFS, MACH, DELTAR, IMPACI, CPSTAG, TFS, CP, G, ISHADI, IFIRST, ISBP, IFSCY, IGTS, IMPS, ISHS, L, XLE, DELTAE, SWEEP, CPNIN, DELCPC, XATACH, XSEPP, ISPNT, IMPACT, ISHAD, RETRAN)

f. Length

10208 bytes

DECK AROK

```
      SUBROUTINE FLOSEP (IGT,LL,N,M,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,
     1         YPA,ZPA,IGTYPE,IORIEN,IPRCK,DELTA,PFS,MACH,DELTAR,IMPACI,
     2         CPSTAG,TFS,CP,G,ISHADI,IFIRST,    ISBP,IFSCY,IGTS,IMPS,
     3         ISHS,L,XLE,DELTAE,SWEEP,CPNIN,DELCPC,XATACH,XSEPP,ISPNT,
     4         IMPACT,ISHAD,RETRAN,NW,TWALL,IRED11)
C
C*****************************************************************
C******   THIS SUBROUTINE DETERMINES THE EFFECT OF VISCOUS FLOW   ********
C******   SEPARATION ON SURFACE PRESSURES                         ********
C*****************************************************************
C
      DIMENSION  TITLE(15),ANGLE(3),FS(8),BS(8),XPA(4),YPA(4),ZPA(4),
     1  TR(10),RE(2),TS(2),PSEVIS(125),PSEIN(125)
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),
     1  YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)
C
      COMMON  CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1  AREA2,IN,IM,K,LS,FS,BS
C
      INTEGER  CASE,PAGE,ERROR
C
      REAL  MACH,NX,NY,NZ,NX2,NY2,NZ2,NXH,NYH,NZH,MACHH,MACHX,MACHX1,MA,
     1 NXS,NYS,NZS,NXF,NYF,NZF,NXFS,NYFS,NZFS
C
C ***** THE FOLLOWING CARD SHOULD BE ICHECK = 1 FOR CHECKOUT ONLY
      ICHECK = 0
      IF (IGTS.EQ.1 .AND. M.EQ.1) ISCT = 0
      ISCT = ISCT + 1
      IF (ISCT .GT. 125) GO TO 150
      IF (ICHECK .EQ. 1)
     1WRITE (6,900) IGTS,IGT,L,LL,N,M,IFSCY,CP,ISCT
  900 FORMAT (1H0,5X,5HIGTS=I2,5H IGT=I2,3H L=I3,4H LL=I3,3H N=I3,
     1  3H M=I3,7H IFSCY=I2,4H CP=E12.5,7H  ISCT=I4)
C
C ***** CHECK FLOSEP CYCLE NUMBER  *****
```

DECK AROK

```
      GO TO (1,20,40), IFSCY
C
C ****************************************************************************
C ***** FIRST FLOSEP CYCLE WITH SHOCK EXPANSION AND FLAP DEFLECTED *****
C ****************************************************************************
C
C  ***** CHECK IF THIS IS HINGE LINE ELEMENT, THE FIRST ELEMENT OF A
C  ***** STRIP, OR THE FIRST FLAP ELEMENT
    1 IF (IGTS.EQ.1 .AND. IGT.EQ.2 .AND. M.NE.1) GO TO 2
      IF (IGTS.EQ.1 .AND. M.EQ.1) LSS = L - 1
      IF (IGTS.EQ.1 .AND. IGT.EQ.2 .AND. M.EQ.1) GO TO 2
      IF (IGTS.EQ.2 .AND. M.EQ.1) GO TO 9
      GO TO 10
C
C  CHECK IF FLOW SEPARATES
    9 CPA2IN = (BS(2)/PH - 1.0)/(G/2.0 *MACHH*MACHH)
C
      IF (ITRANS .EQ. 0) CPAINC = 2.03*(MACHH**2-1.0)**(-0.306)
     1                           /REAHL**0.25
      IF (ITRANS .EQ. 1) CPAINC = 2.2 / REAHL**0.1
C  SET FLOW SEPARATION FLAG ON OR OFF (KFLSP = 1 OR 0)
      KFLSP = 0
      IF (CPA2IN .GT. CPAINC) KFLSP = 1
      IF (RETRAN .LT. 0.0) KFLSP = 0
      DELTER=SQRT((NY*NZH-NZ*NYH)**2+(NZ*NXH-NX*NZH)**2+(NX*NYH-NY*NXH)
     1    **2)
      IF (ABS(DELTER) .GT. 1.0) DELTER = 1.0
      DELTER = ABS(ARSIN(DELTER))
      IF (NX .LT. NXH) DELTER = -DELTER
      IF (ICHECK .EQ. 1)
     1WRITE (6,901) BS(2),MACHH,CPA2IN,REAFT,REAHL,CPAINC,ITRANS,KFLSP
  901 FORMAT (1H ,6E14.5,2I2)
C
      GO TO 10
C
C  SAVE HINGE LINE ELEMENT GEOMETRY DATA
```

FLOSEP

```
DECK AROK

    2  PH = BS(2)                                                        AROK  0720
       TH = BS(3)                                                        AROK  0730
       MACHH = BS(6)                                                     AROK  0740
       NXH = NX                                                          AROK  075C
       NYH = NY                                                          AROK  0760
       NZH = NZ                                                          AROK  0770
       LSEP = 0                                                          AROK  0780
       LATT = 0                                                          AROK  0790
       XHL = (XPA(2) + XPA(3)) / 2.0                                     AROK  0800
       YHL = (YPA(2) + YPA(3)) / 2.0                                     AROK  0810
       ZHL = (ZPA(2) + ZPA(3)) / 2.0                                     AROK  0820
       XSEPP = XHL                                                       AROK  0830
       XATACH = XSEPP                                                    AROK  0840
C  CALCULATE REYNOLDS NUMBER AT THE HINGE LINE                           AROK  0850
       XLEH = XLE -((XPA(2)+XPA(3))/2.0) + XCENT                         AROK  086C
       REAHL = XLEH * BS(8)                                              AROK  087C
       REAFT = BS(8)                                                     AROK  088C
       ITRANS = 0                                                        AROK  0890
       IF (REAHL .GE. RETRAN) ITRANS = 1                                 AROK  0900
C                                                                        AROK  0910
C                                                                        AROK  092C
   10  PSEIN(ISCT) = BS(2)                                               AROK  0930
       PSEVIS(ISCT) = BS(2)                                              AROK  0940
       ISBP = 1                                                          AROK  0950
C                                                                        AROK  0960
C   IS THIS THE LAST ELEMENT IN THE STRIP                                AROK  0970
       IF (IGTS.EQ.2 .AND. IGY.EQ.1) GO TO 14                            AROK  098C
       GO TO 100                                                         AROK  099C
   14  J = IN(N+10)                                                      AROK  1000
       DO 11 I=1,J                                                       AROK  1010
   11  BACKSPACE 3                                                       AROK  102C
       J = IN(4)                                                         AROK  1030
       DO 12 I=1,J                                                       AROK  104C
   12  BACKSPACE 11                                                      AROK  1050
       L = LSS                                                           AROK  1060
       XTE = (XPA(2) + XPA(3)) / 2.0                                     AROK  1070
```

```
DECK AROK

      YTE = (YPA(2) + YPA(3)) / 2.0                                      AROK
      ZTE = (ZPA(2) + ZPA(3)) / 2.0                                      AROK
      CFLAP = SQRT ((XTE-XHL)**2 + (YTE-YHL)**2 + (ZTE-ZHL)**2)          AROK
      IF (ICHECK .EQ. 1)                                                 AROK
     1WRITE (6,903)                                                      AROK
  903 FORMAT (1H ,21HLAST ELEMENT ON STRIP  )                            AROK
C                                                                        AROK
C  IS FLOW SEPARATION FLAG ON                                            AROK
      IF (KFLSP .EQ. 1) GO TO 13                                         AROK
      IFSCY = 3                                                          AROK
      IMPACT = IMPS                                                      AROK
      ISHAD = ISHS                                                       AROK
      DELTAS = DELTAE                                                    AROK
      DELTAE = 0.0                                                       AROK
      GO TO 100                                                          AROK
C                                                                        AROK
   13 IFSCY = 2                                                          AROK
      ISEP = 0                                                           AROK
      GO TO 100                                                          AROK
C                                                                        AROK
C                                                                        AROK
C *********************************************************************  AROK
C ***** SECOND FLOSEP CYCLE.  VISCOUS CALCULATION CYCLE  ***************  AROK
C *********************************************************************  AROK
C                                                                        AROK
C  IS THIS ELEMENT ON FORE SURFACE OR CN FLAP                            AROK
   20 IF (IGTS .EQ. 2) GO TO 30                                          AROK
C                                                                        AROK
      IF (ISEP .EQ. 1) GO TO 29                                          AROK
C *****CALCULATE REQUIRED DATA TO CHECK FOR FLOW SEPARATION POINT        AROK
      TR(5) = TWALL                                                      AROK
      TR(6) = TWALL                                                      AROK
      CALL TEMP (XLE,TR,RE,TS,NW,MER,0,RT)                               AROK
      IF (ITRANS.EC.0) REASFT = RE(1) / XLE                              AROK
      IF (ITRANS.EQ.1) REASFT = RE(2) / XLE                              AROK
      IF (ITRANS.EQ.0) DSQXP = 5.2 / SQRT(REASFT)                        AROK
```

FLOSEP

DECK AROK

```
      IF (ITRANS.EQ.1) DSQXP = 0.154 / REASFT**0.1428*XLE**0.357
      DO = DSQXP * SQRT(XLE)
      IF (ISEP .EQ. 0) GO TO 202
      DXSEP = XLE - XSEP
      IF (XLE .LT. XSEP) GO TO 201
      GO TO 22
C
 202  MA = BS(6)
      REAXOP = XLE * BS(8)
      IF (ITRANS.EQ.0) CPAP=1.56*(MA*MA-1.0)**(-0.262)/REAXOP**0.25
      IF (ITRANS.EQ.1) CPAP=1.91*(MA*MA-1.0)**(-0.309)/REAXOP**0.1
      PPPO = 0.7*MA*MA*CPAP + 1.0
      IF (ITRANS.EQ.0) D1DO =5.69E5*MA**(-4.1)*(PPPO-1.0)**3.5
      IF (ITRANS.EQ.1) D1DO =1.1E6*(MA**(-1.67)*(PPPO-1.0))**8.55
      D1 = D1DO * DO
      XSEP = XLEH - D1
      DXSEP = XLE - XSEP
C
      IF (ICHECK .EQ. 1)
     1WRITE (6,904)
 904  FORMAT (1H0,31HCHECK FOR SEPARATION ON ELEMENT   )
      IF (ICHECK .EQ. 1)
     1WRITE (6,950) REAXOP,CPAP,PPPO,D1DO,REASFT,DSQXP,DO,D1,XSEP,XLE
 950  FORMAT (1H ,5E14.5,/1H ,5E14.5)
C
C ***** CHECK IF SEPARATION POINT HAS BEEN REACHED
      IF (XLE .GE. XSEP) GO TO 22
C     SEPARATION POINT HAS NOT BEEN REACHED
      ISEP = 0
 201  DXSEP1 = DXSEP
      XLE1 = XLE
      MACHX1 = BS(6)
      PX1 = BS(2)
      TX1 = BS(3)
      PPPO1 = PPPO
      DO1 = DO
```

DECK AROK

```
      CPSEP = CP
      GO TO 21
C
C     SEPARATION POINT HAS BEEN REACHED
   22 LSEP = ISCT
C
C ***** CALCULATE CONDITIONS AT EXACT SEPARATION POINT
      IF (IGTS.EQ.1 .AND. M.EQ.1) GO TO 31
      XLESEP = XLE1 - DXSEP1*(XLE-XLE1)/(DXSEP-DXSEP1)
      IF (ICHECK .EQ. 1) WRITE (6,909) XLESEP,XLE1,DXSEP1,XLE,XLE1,
     1  DXSEP,DXSEP1
  909 FORMAT (1H ,7E14.5)
      IF (ISEP .EQ. 0) D1 = XLEH - XLESEP
      XSEPP = XSEPP + D1
      MACHX = (MA-MACHX1)/(XLE-XLE1)*(XLESEP-XLE1) + MACHX1
      PPPOX = (PPPO-PPPO1)/(XLE-XLE1)*(XLESEP-XLE1) + PPPO1
      PX = (BS(2) - PX1)/(XLE-XLE1)*(XLESEP-XLE1)+PX1
      CPX = (PX/PFS-1.0)/(G/2.0*MACH*MACH)
      GO TO 19
C  SET UP DATA  FLOW SEPARATED AT LEADING EDGE
   31 D1 = XLEH
      ELFI = 0.0
      PX = BS(2)
      MACHX = BS(6)
      XLESEP = 0.0
      XSEPP = (XPA(1) + XPA(4)) / 2.0
   19 CFD1 = CFLAP / D1
      EMADF = MACHX * DELTER
      IF (ITRANS .EQ. 0) D2D1 = (0.545 - 0.04*EMADF) * SQRT(CFD1)
      IF (ITRANS .EQ. 1) D2D1 = (1.16  - 0.33*EMADF) * SQRT(CFD1)
      IF (CFD1 .GE. 1.0) GO TO 24
      IF (CFD1 .GT. 0.25) GO TO 23
      IF (ITRANS .EQ. 0) D2D1 = 0.273 - 0.02*EMADF
      IF (ITRANS .EQ. 1) D2D1 = 0.58 - 0.165*EMADF
      ELAM = 0.25
      GO TO 25
```

FLOSEP

FLOSEP

DECK AROK

```
   23  ELAM = CFD1                                                        AROK  2160
       GO TO 25                                                           AROK  2170
   24  IF (ITRANS .EQ. 0) D2D1 = 0.545 - 0.04*EMADF                       AROK  2180
       IF (ITRANS .EQ. 1) D2D1 = 1.16 - 0.33*EMADF                        AROK  2190
       ELAM = 1.0                                                         AROK  2200
   25  IF ((ITRANS.EQ.0 .AND. EMADF.LT.5.0) .OR.                          AROK  2210
      1    (ITRANS.EQ.1 .AND. EMADF.LT.2.4)) GO TO 26                     AROK  2220
       IF (ITRANS .EQ. 0) D2D1 = 0.344 * SQRT(ELAM)                       AROK  2230
       IF (ITRANS .EQ. 1) D2D1 = 0.37 * SQRT(ELAM)                        AROK  2240
   26  D2 = D2D1 * D1                                                     AROK  2250
       IF (ISEP .EQ. 2) D2 = D3                                           AROK  2260
       XATACH = XATACH - D2                                               AROK  2270
C                                                                         AROK  2280
C                                                                         AROK  2290
C   CALCULATE DOWNSTREAM INTERACTION LENGTH TO PRESSURE RISE              AROK  2300
       IF (IGTS.EQ.1 .AND. M.EQ.1) GO TO 17                               AROK  2310
       IF (ISEP .EQ. 2) GO TO 200                                         AROK  2320
       SINTH = SQRT(CPAP*(G+1.0)/4.0+1.0/(MACHX*MACHX))                   AROK  2330
       THETA = ARSIN(SINTH)                                               AROK  2340
       TANPHI = 1.0 / ((2.0/CPAP-1.0)*SINTH/COS(THETA))                   AROK  2350
       GO TO 18                                                           AROK  2360
   17  TANPHI = D2*SIN(DELTER) / (D1 + D2*COS(DELTER))                    AROK  2370
   18  PHI = ATAN(TANPHI) * 57.29578                                      AROK  2380
       D3D1 = (TANPHI/(SIN(DELTER)/COS(DELTER)-TANPHI))/COS(DELTER)       AROK  2390
       D3 = D3D1 * D1                                                     AROK  2400
       IF (D3 .GT. D2) D3 = D2                                            AROK  2410
       IF (D3 .LE. CFLAP) GO TO 200                                       AROK  2420
       D3 = CFLAP                                                         AROK  2430
       D1 = D3 / D3D1                                                     AROK  2440
       XSEPP = XHL                                                        AROK  2450
       XSEP = XLEH - D1                                                   AROK  2460
       DXSEP = XLE1 - XLE                                                 AROK  2470
       ISEP = 2                                                           AROK  2480
       LSEP = 0                                                           AROK  2490
       GO TO 201                                                          AROK  2500
C                                                                         AROK  2510
```

```
DECK AROK

C   CALCULATE PLATEAU PRESSURE AND PEAK FLAP PRESSURE                   AROK  2520
 200  ISEP = 1                                                          AROK  2530
      FS2 = FS(2)                                                       AROK  2540
      FS6 = FS(6)                                                       AROK  2550
      FS(2) = PX                                                        AROK  2560
      FS(6) = MACHX                                                     AROK  2570
      ANGLE(2) = PHI                                                    AROK  2580
      ISDET = 0                                                         AROK  2590
      IPRCK = 0                                                         AROK  2600
      IFIRST = 0                                                        AROK  2610
      CALL COMPR (ANGLE,MER,IPRCK,CPSTAG,TFS,PFS,ISDET,IFIRST,DUMMY)    AROK  2620
      IF (IGTS.EQ.1 .AND. M.EQ.1) GO TO 203                             AROK  2630
C                                                                       AROK  2640
C  CALCULATE FREE INTERACTION LENGTH                                    AROK  2650
      PPPOX = BS(2) / FS(2)                                             AROK  2660
      IF (ITRANS.EQ.0) ELFIDO = 2.47E5*MACHX**(-4.2)*(PPPOX-1.0)**3.45  AROK  2670
      IF (ITRANS.EQ.1) ELFIDO = 1.84E4*((PPPOX-1.0)/MACHX**1.325)       AROK  2680
     1                        **(-8.4)                                  AROK  2690
      DOX = (DO-DO1)/(XLE-XLE1)*(XLESEP-XLE1) + DO1                     AROK  2700
C                                                                       AROK  2710
C   CALCULATE DOWNSTREAM INTERACTION LENGTH TO PEAK PRESSURE            AROK  2720
      ELFI = ELFIDO * DOX                                               AROK  2730
C                                                                       AROK  2740
C ***** PLATEAU PRESSURE *************                                  AROK  2750
 203  CPIP = (BS(2)/PFS - 1.0) / (G/2.0*MACH*MACH)                      AROK  2760
      FS(2) = BS(2)                                                     AROK  2770
      FS(6) = BS(6)                                                     AROK  2780
      ANGLE(2) = DELTER*57.29578 - PHI                                  AROK  2790
      CALL COMPR (ANGLE,MER,IPRCK,CPSTAG,TFS,PFS,ISDET,IFIRST,DUMMY)    AROK  2800
C                                                                       AROK  2810
C ***** PEAK FLAP PRESSURE *****                                        AROK  2820
      CPI2 = (BS(2)/PFS - 1.0) / (G/2.0 *MACH*MACH)                     AROK  2830
      FS(2) = FS2                                                       AROK  2840
      FS(6) = FS6                                                       AROK  2850
C                                                                       AROK  2860
C                                                                       AROK  2870
```

FLOSEP

DECK AROK.

```
C    CALCULATE VISCOUS PRESSURE COEFFICIENT WITH SEPARATION
  29  CPSEP = CPIP
      IF (XLE.GE.XLESEP .AND. XLE.LT.(XLESEP+ELFI))
     1  CPSEP = ((CPIP-CPX)/ELFI)*XLE + CPX - (CPIP-CPX)/ELFI * XLESEP
      IF (ICHECK .EQ. 1)
     1WRITE (6,960) XSEP,D1,D2,D3,ELFI,CPIP,CPI2
 960  FORMAT (1H ,21HSEPARATION CONDITIONS,/1H ,6H XSEP=F8.3,4H D1=F8.3,
     1 4H D2=F8.3,4H D3=F8.3,6H ELFI=F8.3,6H CPIP=E12.5,6H CPI2=E12.5 )
      IF (ICHECK .EQ. 1)
     1WRITE (6,951) D2D1,EMADF,CFD1,ELFI,DOX,ELFIDO,PPPDX,XLESEP,
     2  DXSEP,XSEP,D2,PHI,D3D1,D3
 951  FORMAT (1H ,8E14.5,/1H ,6E14.5)
      GO TO 21
C
C
C   CALCULATE CP VISCOUS ON THE FLAP
  30  IF (ISEP .EQ. 1) GO TØ 32
      CPSEP = CP
      GO TO 28
  32  IF ((XLE.GT.(XLEH+D3)) .AND. (XLE.LT.(XLEH+D2))) GO TO 33
      IF (XLE .GE. (XLEH+D2)) GO TO 27
      CPSEP = CPIP
      GO TO 28
  27  CPSEP = CPI2
      IF (LATT .EQ. 0) LATT = ISCT
      GO TO 28
  33  CPSEP = (CPIZ-CPIP)/(D2-D3)*(XLE-XLEH-D3) + CPIP
      IF (D2 .GT. CFLAP) CPSEP = CPIP
C
CC  IS THIS THE LAST STRIP ELEMENT
  28  IF (IGTS.EQ.2 .AND. IGT.EQ.1) GO TO 34
      GO TO 21
C
  34  J = IN(N+10)
      DO 35 I=1,J
  35  BACKSPACE 3
```

```
DECK AROK
      J = IN(4)
      DO 36 I=1,J
   36 BACKSPACE 11
C
      IF (ICHECK .EQ. 1)
     1WRITE (6,906)
  906 FORMAT (1H ,34HLAST STRIP ELEMENT ON SECOND CYCLE  )
      L = LSS
      IFSCY = 3
      IMPACT = IMPS
      ISHAD = ISHS
      DELTAS = DELTAE
      DELTAE = 0.0
C
C
C ***** CALCULATE PRESSURE COEFFICIENT INCREMENT DUE TO SEPARATION
   21 PSEVIS(ISCT) = (CPSEP*G/2.0 * MACH*MACH + 1.0) * PFS
      IF (ICHECK .EQ. 1)
     1WRITE (6,970) CPSEP,PSEVIS(ISCT),ISCT
  970 FORMAT (1H ,5X,2E14.5,I5)
      GO TO 100
C
C
C***** THIRD FLOSEP CYCLE  *****
C
   40 ISPNT = 0
      PNIN = (CP*G/2.0 * MACH*MACH + 1.0) * PFS
C  IS THIS ELEMENT ON FORE SURFACE OR ON THE FLAP
      IF (IGTS .EQ. 2) GO TO 50
      IF (ISCT .EQ. LSEP) ISPNT = 1
      BS2F = PSEIN(ISCT)
      IF (ICHECK .EQ. 1)
     1WRITE (6,907)
  907 FORMAT (1H0,25H3RD CYCLE ON FORE-SURFACE  )
      GO TO 80
   50 BS2 = BS(2)
```

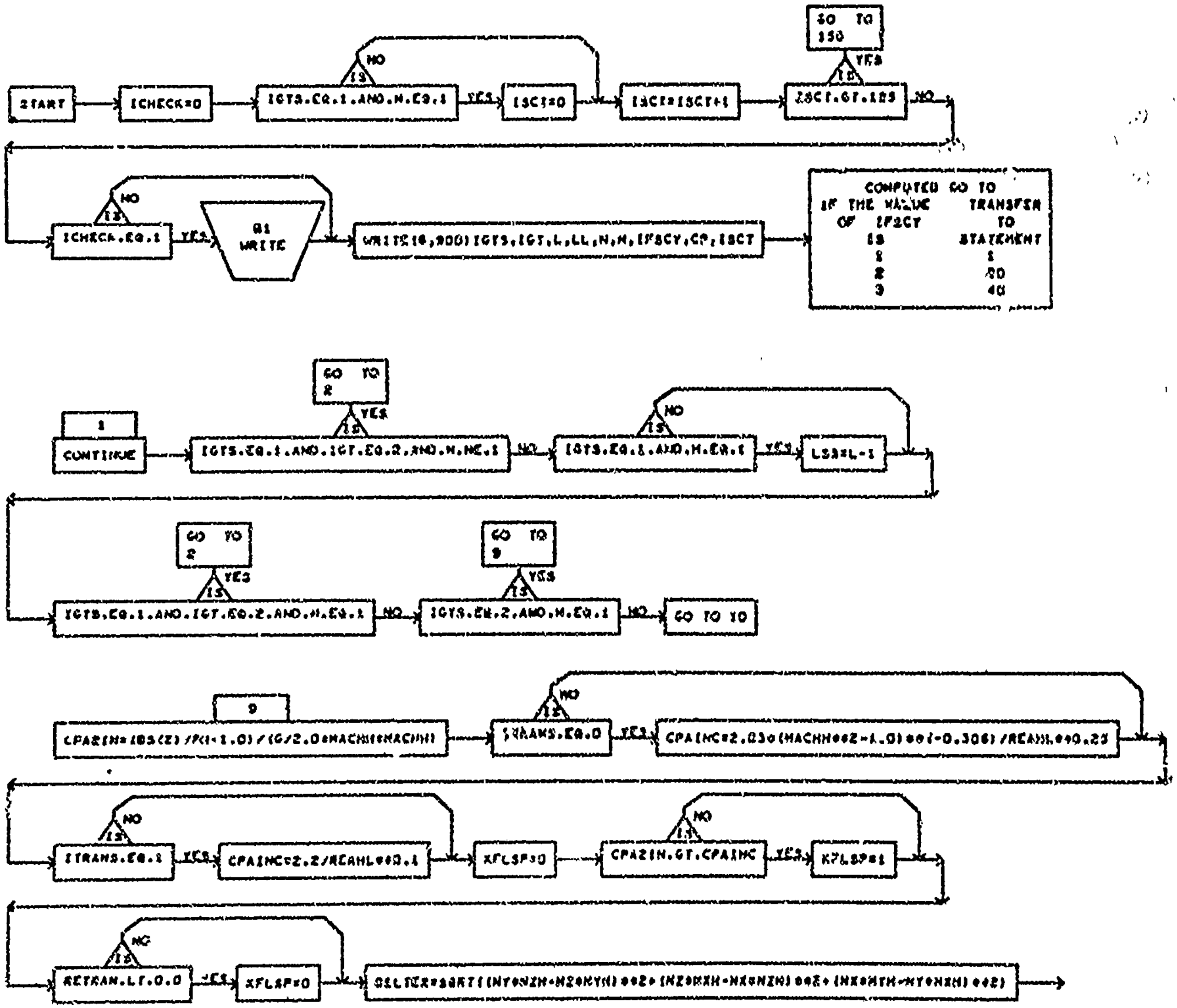
```
DECK AROK
      BS3 = BS(3)
      BS6 = BS(6)
      NXS = NX
      NYS = NY
      NZS = NZ
      IF (IGTS.EQ.2 .AND. M.EQ.1) GO TO 60
      BS(2) = BS2F
      BS(3) = BS3F
      BS(6) = BS6F
      NXF = NXFS
      NYF = NYFS
      NZF = NZFS
      GO TO 61
C
   60 BS(2) = PH
      BS(3) = TH
      BS(6) = MACHH
      NXF = NXH
      NYF = NYH
      NZF = NZH
C
C
   61 CALL SHKEXP (1,1,1,0,NXF,NYF,NZF,0.0,PFS,MACH,1,
     1  0.0,1,0.0,0.0,CPF,0.0,1,0,0,0,1,0.0,0.0,0.0,0.0,0.0,1)
      CALL SHKEXP (1,N,M,0,NX,NY,NZ,DELTA,PFS,MACH,IGTS,
     1  DELTAR,IMPACI,CPSTAG,TFS,CPF,G,ISHADI,0,LL,0,IGTYPE,
     2  1.0,1.0,1.C,1.0,1.0,0)
C
      BS2F = BS(2)
      BS3F = BS(3)
      BS6F = BS(6)
      NXFS = NX
      NYFS = NY
      NZFS = NZ
C  SET SHOCK EXPANSION DATA BACK TO THE ORIGINAL CONDITIONS
      BS(2) = BS2
```

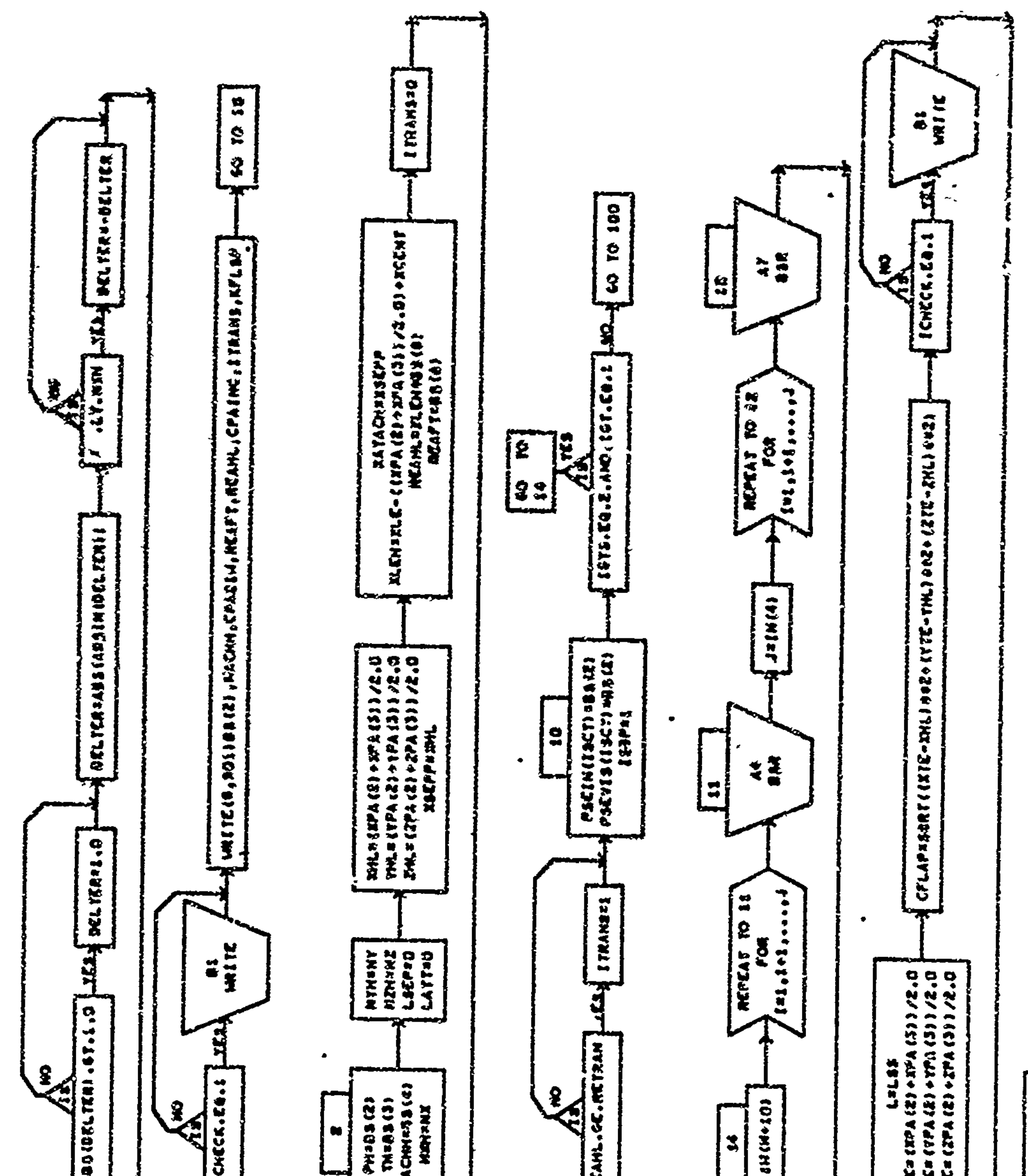
```
DECK AROK
      BS(3) = BS3
      BS(6) = BS6
      NX = NXS
      NY = NYS
      NZ = NZS
      CALL SHKEXP (1,1,1,0,NX ,NY ,NZ ,0.0,PFS,MACH,1,
     1  0.0,1,0.0,0.0,CPF,0.0,1,0,0,0,1,0.0,0.0,0.0,0.0,0.0,1)
      READ (11)  LL,N,M,NX,NY,NZ,XCENT,YCENT,ZCENT,AREA,XPA,YPA,ZPA,XLE
      IREDI1 = 1
      IF (ISCT .EQ. LATT) ISPNT = 2
C
C
C ***** CALCULATE FINAL DELTA PRESSURE TO BE APPLIED TO NORMAL CP
   80 DELP = (PSEVIS(ISCT) - BS2F) / BS2F*PNIN
      P = PNIN + DELP
      CPNIN = (PNIN/PFS - 1.0) / (G/2.0 * MACH*MACH)
      CP = (P/PFS - 1.0) / (G/2.0 * MACH*MACH)
      DELCPC = CP - CPNIN
      IF (ABS(DELCPC) .LT. 1.0E-06) DELCPC = 0.0
      IF (ICHECK .EQ. 1)
     1WRITE (6,800) ISCT,CP,P,DELP,PSEVIS(ISCT),BS2F,PNIN
  800 FORMAT (1H ,12HFINAL RESULT, 6H ISCT=I3,4H CP=E12.5,3H P=E12.5,
     1 6H DELP=E12.5,8H PSEVIS=E12.5,6H BS2F=E12.5,6H PNIN=E12.5)
C
      ISBP = 0
C
      IF (IGTS.EQ.2 .AND. IGT.EQ.1) GO TO 81
      GO TO 100
   81 DELTAE = ..LTAS
      IF (L .EQ. LL) GO TO 100
      IFSCY = 1
      IMPACT = 9
      ISHAD = 7
  100 RETURN
  150 WRITE (6,151)
  151 FORMAT (1H ,46H**** NUMBER OF STREAMWISE ELEMENTS FOR CONTROL
```

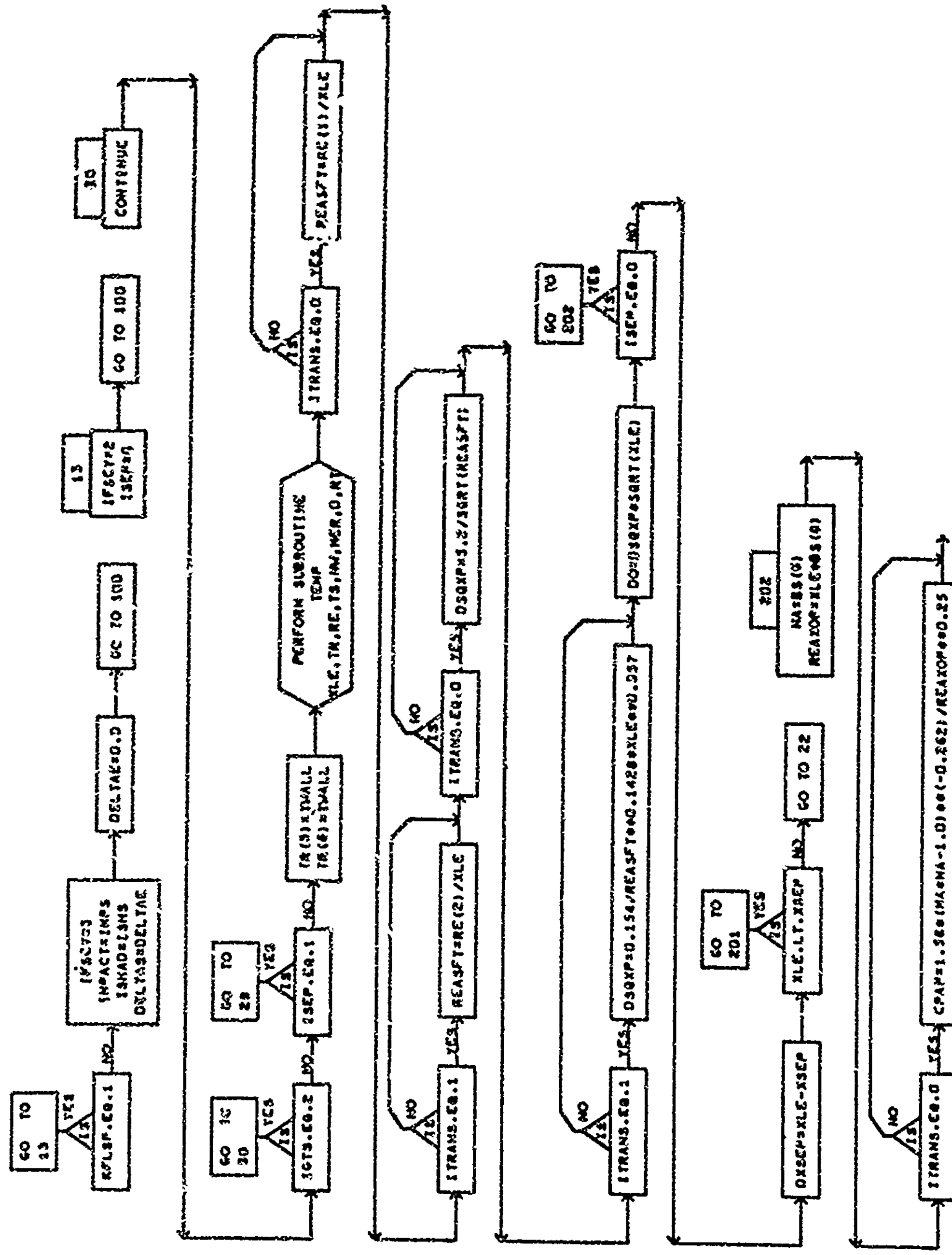
DECK AROK

```
  1 45H FORE-SURFACE AND FLAP CANNOT EXCEED 12% ****)          AROK  4320
   ERROR = 1                                                   AROK  4330
   RETURN                                                      AROK  4340
   END                                                         AROK  4350
```

SUBROUTINE FLOSEP



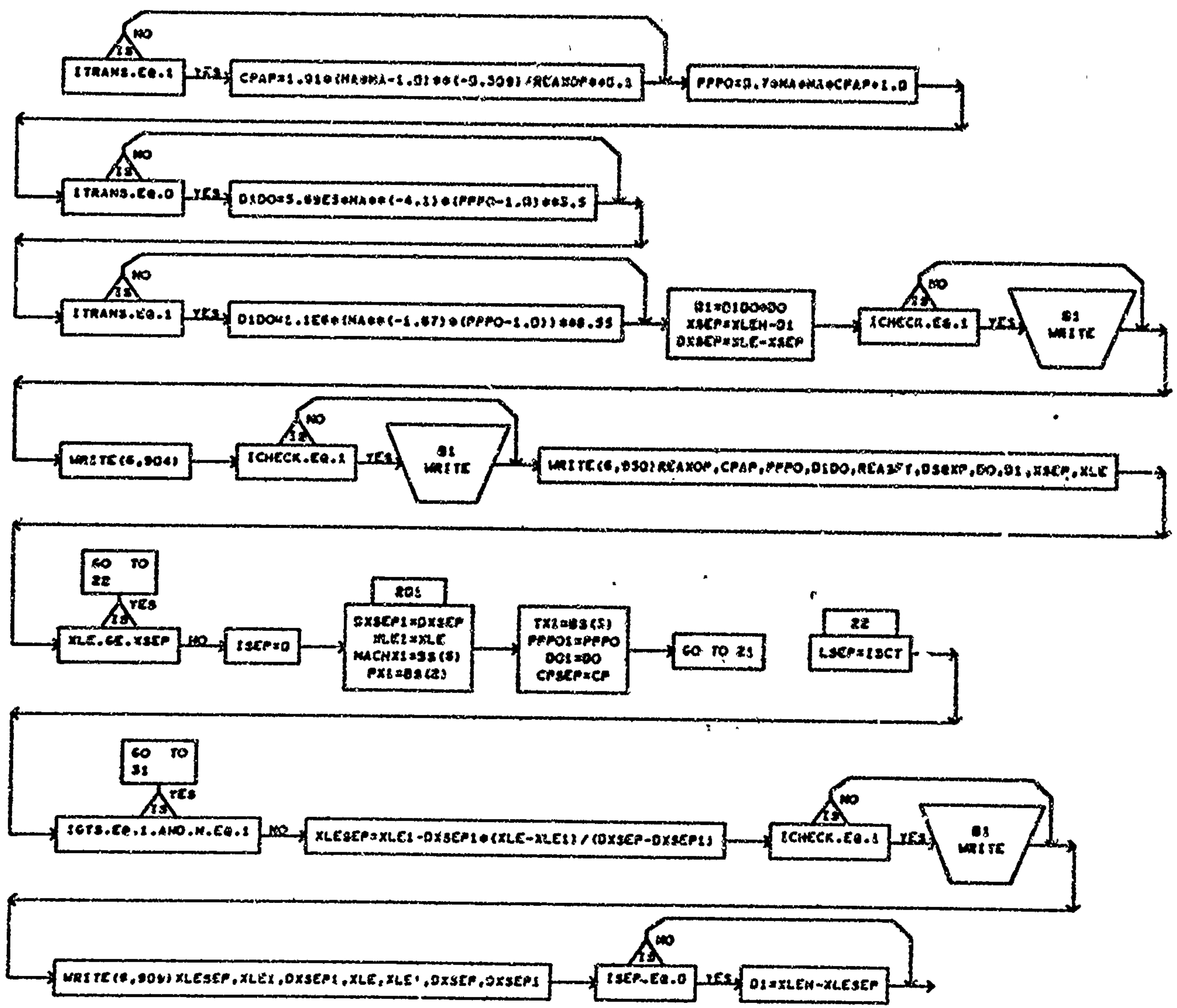


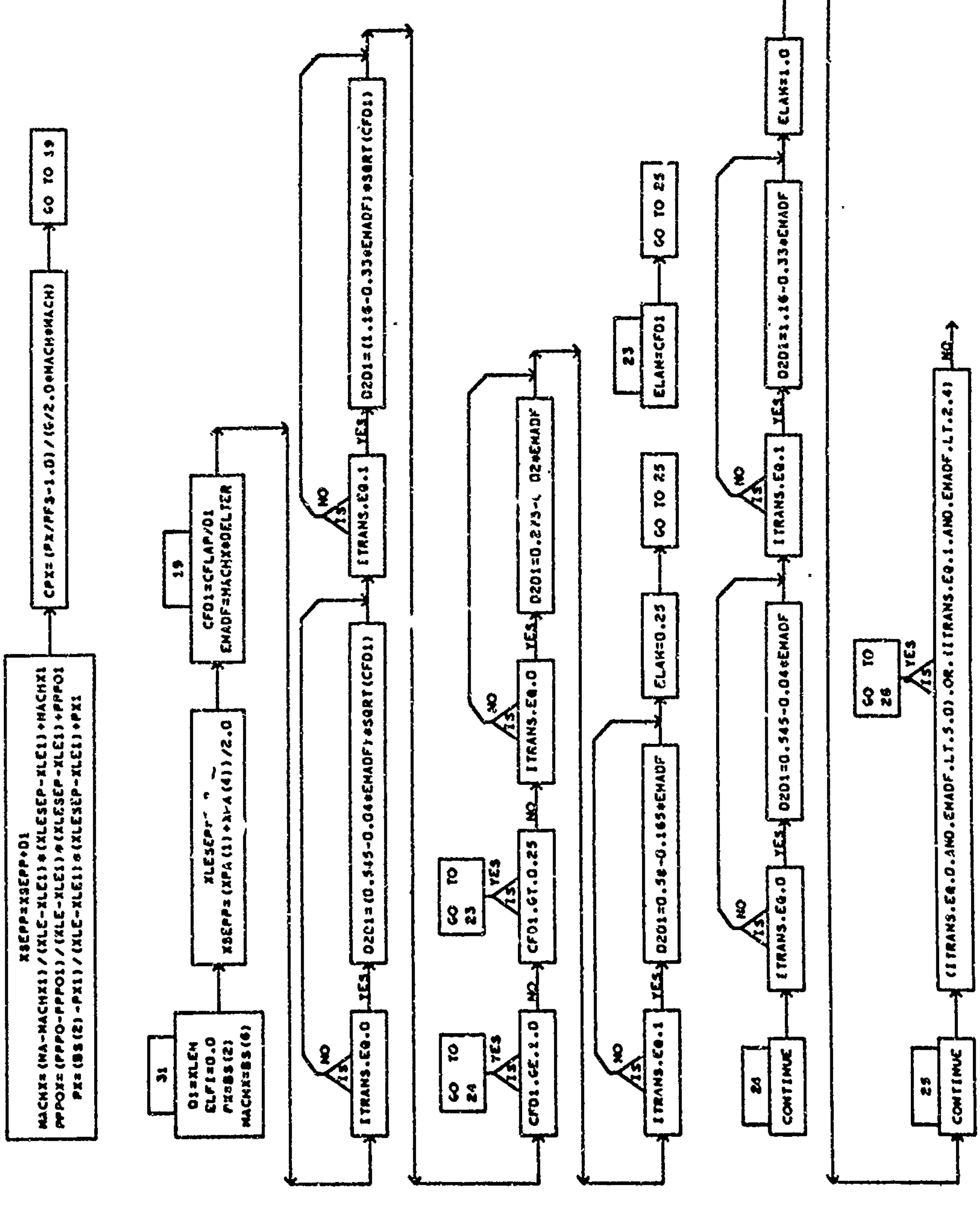


GO TO 13
KFLSF.EQ.1
DELTAE=0.0
GO TO 100
GO TO 100
30 CONTINUE
GO TO 30
GO TO 28
ISEP.EQ.1
PERFORM SUBROUTINE TEMP
XLE,TR,RE,TS,MA,MCR,D,KT
ITRANS.EQ.0
REASFT=RE(1)/XLE
ITRANS.EQ.1
REASFT=RE(2)/XLE
DSQXP=5.2/SQRT(REASFT)
DSQXP=0.154/REASFT**0.1428*XLE**0.357
DO=DSQXP*SQRT(XLE)
ISEP.EQ.0
GO TO 202
DXSEP=XLE-XSEP
XLE.LT.XREP
GO TO 201
GO TO 22
202
MA=BS(0)
REAXOP=XLE*BS(0)
CPAP=1.58*(MA*MA-1.0)**(-0.262)/REAXOP**0.25

FLOSEP

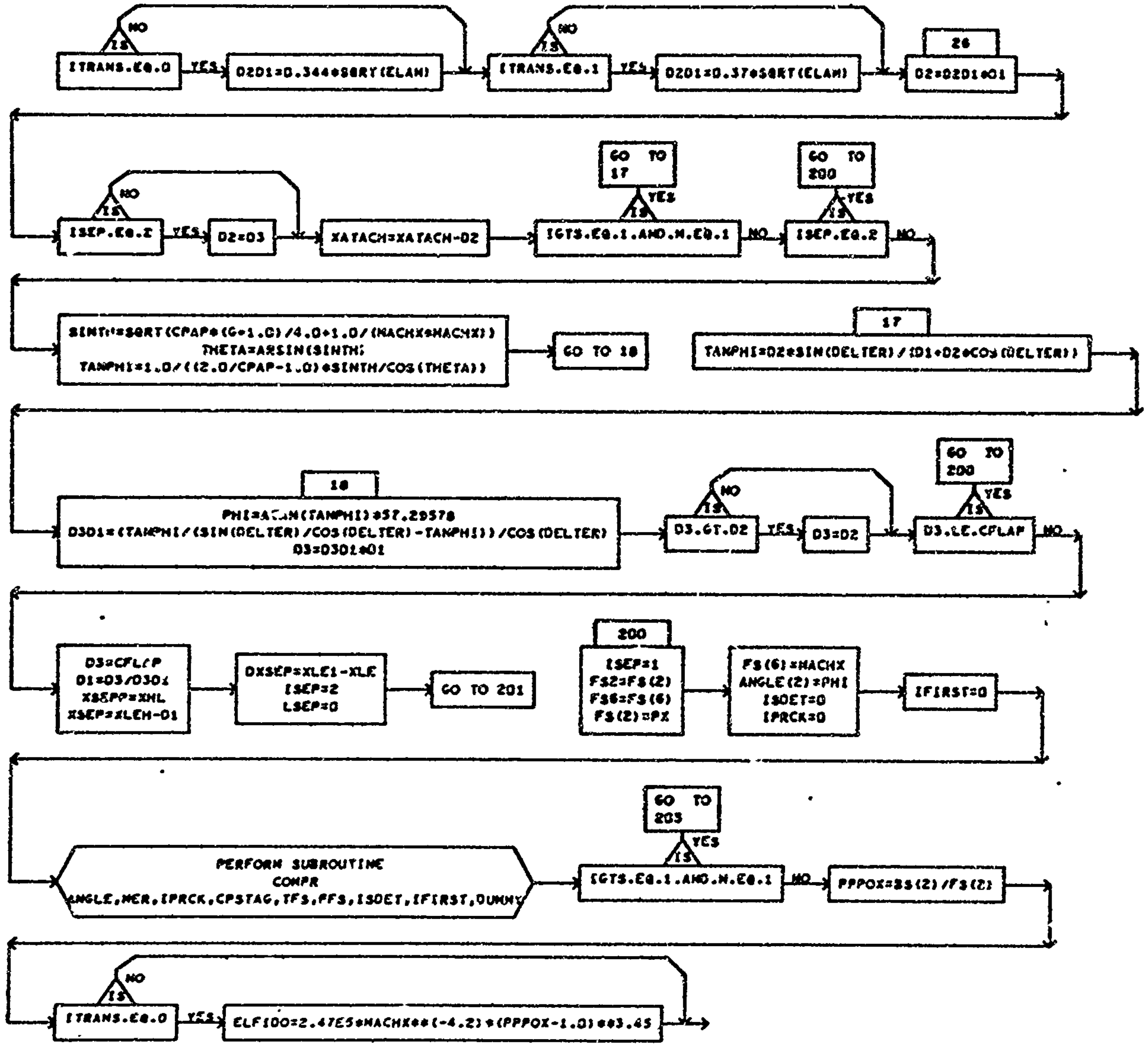
FILOSEP

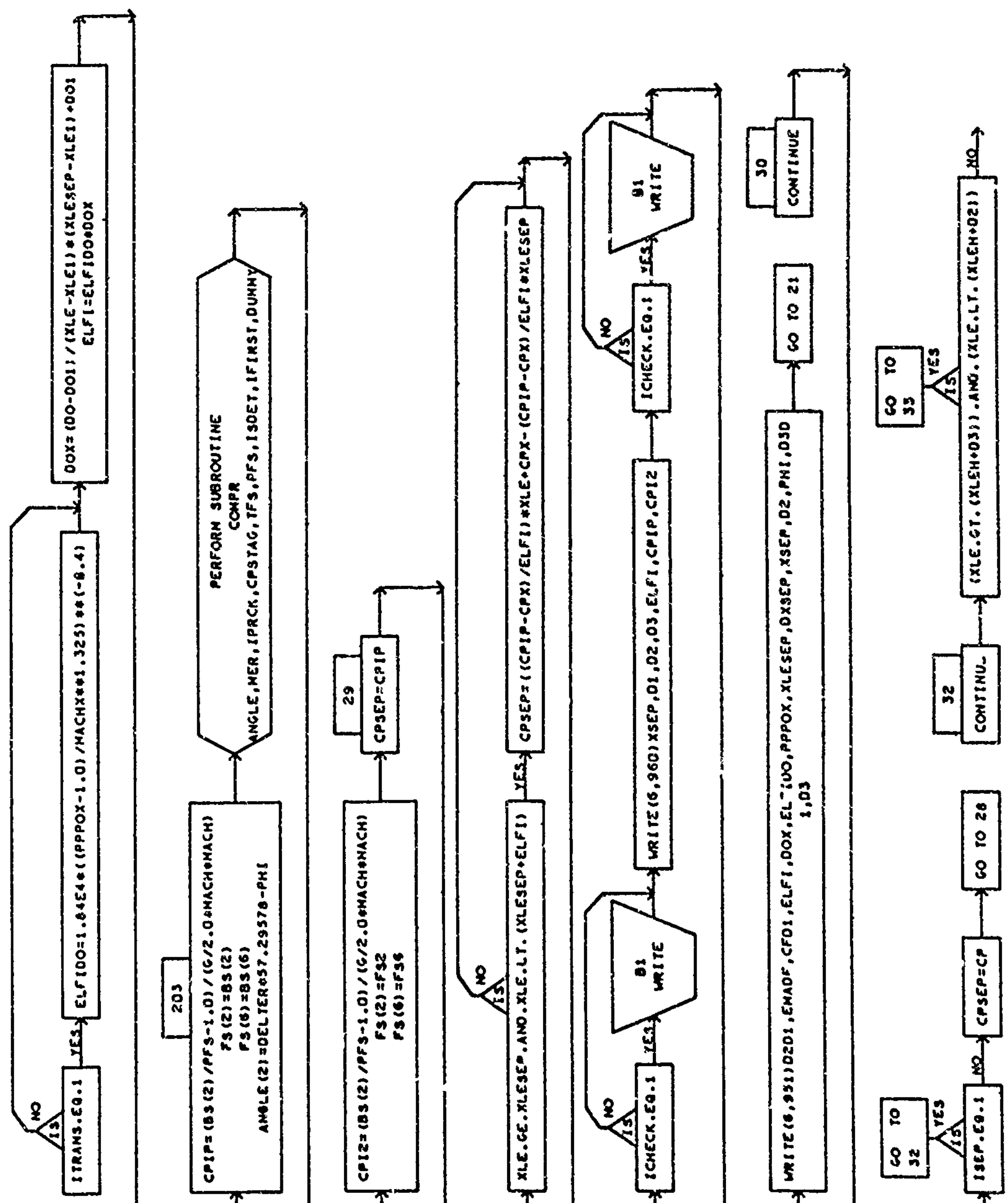

ITRANS.EQ.1
IS
NO
YES
ITRANS.EQ.0
ICHECK.EQ.1
B1
WRITE
WRITE(6,904)
XLE.GE.XSEP
GO TO
22
ISEP=0
201
GO TO 21
LSEP=ISCT
31
ISEP.EQ.0
D1=XLEN-XLESEP

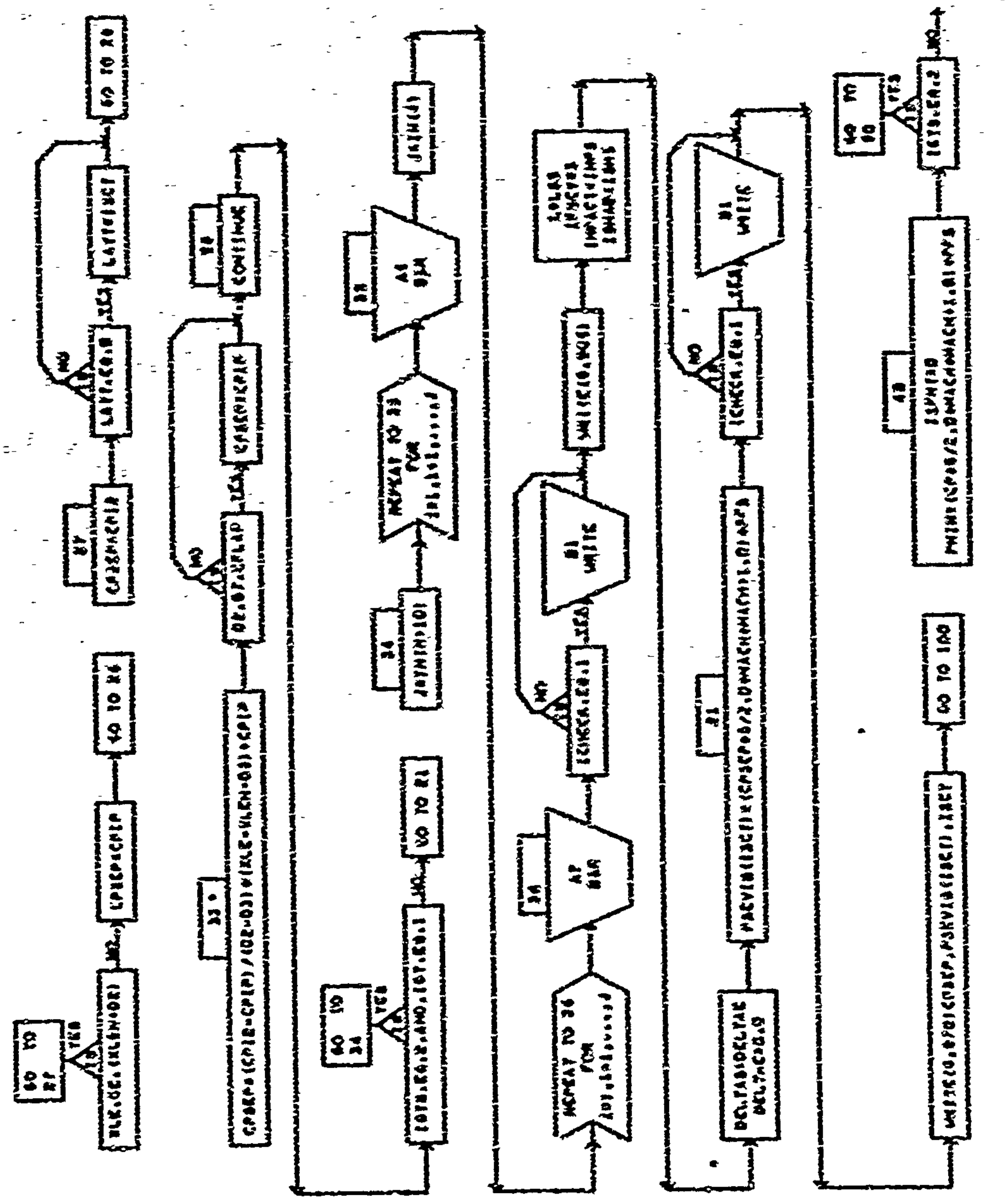

XSEPP=XSEPP+D1
MACHX=(MA-MACHX1)/(XLE-XLE1)*(XLESEP-XLE1)+MACHX1
PPPOX=(PPPO-PPPO1)/(XLE-XLE1)*(XLESEP-XLE1)+PPPO1
PX=(BS(2)-PX1)/(XLE-XLE1)*(XLESEP-XLE1)+PX1
CPX=(PX/PFS-1.0)/(G/2.0*MACH*MACH)
GO TO 19
31
D1=XLEM
ELFI=0.0
PX=BS(2)
MACHX=BS(6)
XLESEP
XSEPP=(XPA(1)+XPA(4))/2.0
19
CFD1=CFLAP/D1
EMADF=MACHX*DELTER
IS
ITRANS.EQ.0
NO
YES
D2D1=(0.545-0.04*EMADF)*SQRT(CFD1)
IS
ITRANS.EQ.1
NO
YES
D2D1=(1.16-0.33*EMADF)*SQRT(CFD1)
IS
CFD1.GE.2.0
YES
GO TO 24
NO
IS
CFD1.GT.0.25
YES
GO TO 23
NO
IS
ITRANS.EQ.0
NO
YES
D2D1=0.275-
D2*EMADF
23
ELAM=CFD1
GO TO 25
IS
ITRANS.EQ.1
NO
YES
D2D1=0.58-0.165*EMADF
ELAM=0.25
GO TO 25
26
CONTINUE
IS
ITRANS.EQ.0
NO
YES
D2D1=0.545-0.04*EMADF
IS
ITRANS.EQ.1
NO
YES
D2D1=1.16-0.33*EMADF
ELAM=1.0
25
CONTINUE
IS
(ITRANS.EQ.0.AND.EMADF.LT.5.0).OR.(ITRANS.EQ.1.AND.EMADF.LT.2.4)
YES
GO TO 26
NO

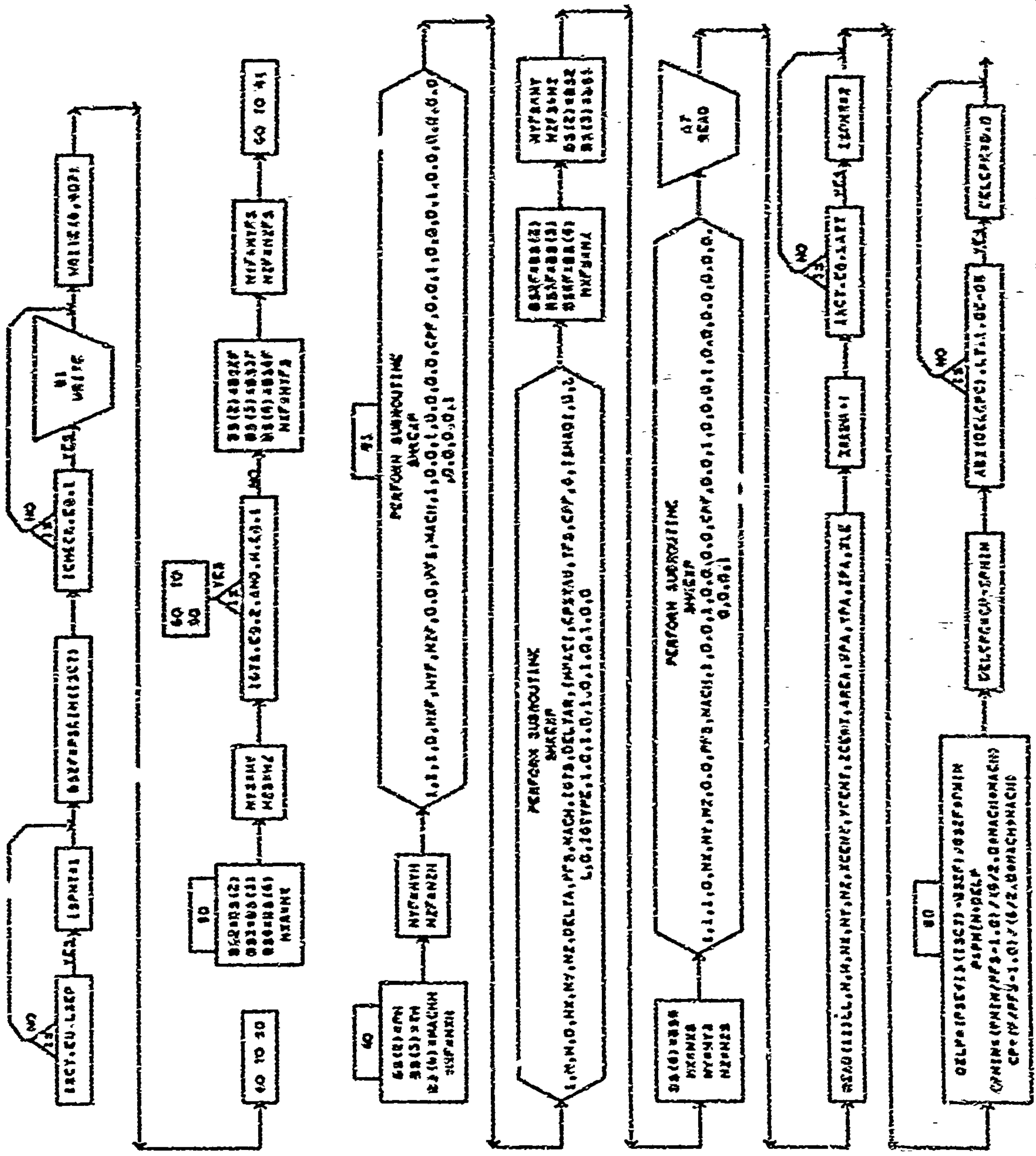
FLOSEP

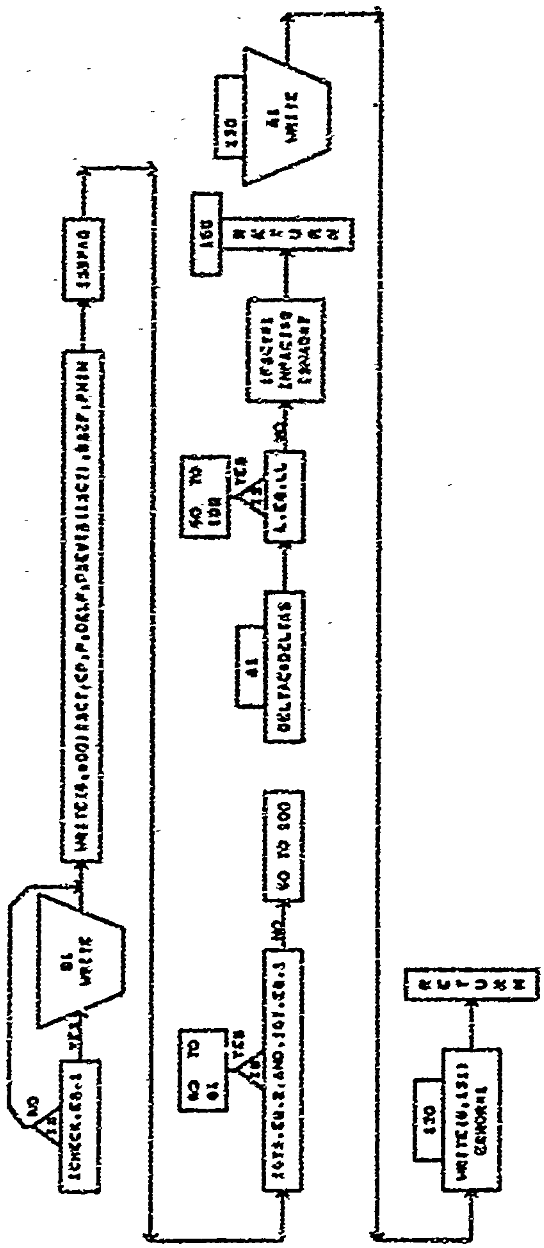
FLOSEP


ITRANS.EQ.0
IS
NO
YES
D2D1=0.344*SQRT(ELAM)
ITRANS.EQ.1
D2D1=0.37*SQRT(ELAM)
26
D2=D2D1*D1
ISEP.EQ.2
D2=D3
XATACH=XATACH-D2
IGTS.EQ.1.AND.M.EQ.1
GO TO 17
ISEP.EQ.2
GO TO 200
SINTH=SQRT(CPAP*(G+1.0)/4.0+1.0/(MACHX*MACHX))
THETA=ARSIN(SINTH)
TANPHI=1.0/((2.0/CPAP-1.0)*SINTH/COS(THETA))
GO TO 18
17
TANPHI=D2*SIN(DELTER)/(D1+D2*COS(DELTER))
18
PHI=ATAN(TANPHI)*57.29578
D3D1=(TANPHI/(SIN(DELTER)/COS(DELTER)-TANPHI))/COS(DELTER)
D3=D3D1*D1
D3.GT.D2
D3=D2
D3.LE.CFLAP
GO TO 200
D3=CFLAP
D1=D3/D3D1
XSEPP=XHL
XSEP=XLEH-D1
DXSEP=XLE1-XLE
ISEP=2
LSEP=0
GO TO 201
200
ISEP=1
FS2=FS(2)
FS6=FS(6)
FS(2)=PX
FS(6)=MACHX
ANGLE(2)=PHI
ISDET=0
IPRCK=0
IFIRST=0
PERFORM SUBROUTINE
COMPR
ANGLE,MER,IPRCK,CPSTAG,TFS,FFS,ISDET,IFIRST,DUMMY
IGTS.EQ.1.AND.M.EQ.1
GO TO 203
PPPOX=SS(2)/FS(2)
ITRANS.EQ.0
ELF100=2.47E5*MACHX**(-4.2)*(PPPOX-1.0)**3.45


ITRANS.EQ.1
IS
NO
YES
ELFIDO=1.64E4*((PPPOX-1.0)/MACHX**1.325)**(-0.4)
DOX=(DO-DO1)/(XLE-XLE1)*(XLESEP-XLE1)+DO1
ELFI=ELFIDO*DOX
203
CPIP=(BS(2)/PFS-1.0)/(G/2.0*MACH*MACH)
FS(2)=BS(2)
FS(6)=BS(6)
ANGLE(2)=DELTER*57.29578-PHI
PERFORM SUBROUTINE
COMPR
ANGLE,MER,IPACK,CPSTAG,TFS,PFS,ISDET,IFIRST,DUMMY
CPI2=(BS(2)/PFS-1.0)/(G/2.0*MACH*MACH)
FS(2)=FS2
FS(6)=FS6
29
CPSEP=CPIP
XLE.GE.XLESEP.AND.XLE.LT.(XLESEP+ELFI)
CPSEP=((CPIP-CPX)/ELFI)*XLE+CPX-(CPIP-CPX)/ELFI*XLESEP
ICHECK.EQ.1
81
WRITE
WRITE(6,960)XSEP,D1,D2,D3,ELFI,CPIP,CPI2
ICHECK.EQ.1
91
WRITE
WRITE(6,951)D2D1,EMADF,CFD1,ELFI,DOX,EL~DO,PPPOX,XLESEP,DXSEP,XSEP,D2,PHI,D3D
1,D3
GO TO 21
30
CONTINUE
ISEP.EQ.1
GO TO 32
CPSEP=CP
GO TO 28
32
CONTINU_
(XLE.GT.(XLEH+D3)).AND.(XLE.LT.(XLEH+D2))
GO TO 33







SYMBOLS USED IN SUBROUTINE FLOSEP

ANGLE	R	D	FLOW ANGLE ARRAY	FLOSEP
AREA	R	A	ELEMENT AREA	FLOSEP
AREA2	R	C	(NOT USED)	FLOSEP
BS	R	C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	FLOSEP
BS2	R	U	PRESSURE TO BE SAVED	FLOSEP
BS2F	R	U	INVISCID SHOCK-EXPANSION PRESSURE	FLOSEP
BS3	R	U	TEMPERATURE TO BE SAVED	FLOSEP
BS3F	R	U	TEMPERATURE TO BE SAVED	FLOSEP
BS6	R	U	MACH NUMBER TO BE SAVED	FLOSEP
BS6F	R	U	MACH NUMBER ON FLAP TO BE SAVED	FLOSEP
CASE	I	C	CASE NUMBER	FLOSEP
CFD1	R	U	FLAP CHORD/UPSTREAM INTERACTION LENGTH	FLOSEP
CFLAP	R	U	FLAP CHORD	FLOSEP
CP	R	A	PRESSURE COEFFICIENT	FLOSEP
CPAINC	R	U	PRESSURE RISE TO CAUSE INCIPIENT SEPARATION	FLOSEP
CPAP	R	U	PLATEAU PRESSURE COEFFICIENT	FLOSEP
CPA2IN	R	U	INVISCID PRESSURE RISE COEFFICIENT ON TO CONTROL SURFACE	FLOSEP
CPF	R	U	PRESSURE COEFFICIENT ON FLAP (DUMMY-NOT USED)	FLOSEP
CPIP	R	U	PLATEAU PRESSURE COEFFICIENT	FLOSEP
CPI2	R	U	PEAK PRESSURE COEFFICIENT ON FLAP	FLOSEP
CPNIN	R	A	PRESSURE COEFFICIENT (INPUT FORCE METHOD, INVISCID)	FLOSEP
CPSEP	R	U	VISCOUS PRESSURE COEFFICIENT WITH SEPARATION	FLOSEP
CPSTAG	R	A	NEWTONIAN CORRELATION FACTOR, K (STAGNATION PRESSURE COEFF)	FLOSEP
CPX	R	U	PRESSURE COEFFICIENT AT SEPARATION POINT	FLOSEP
DELCPC	R	A	PRESSURE COEFFICIENT INCREMENT DUE TO CONTROL SURFACE	FLOSEP
DELP	R	U	PRESSURE INCREMENT DUE TO FLAP AND SEPARATION	FLOSEP
DELTA	R	A	ELEMENT IMPACT ANGLE (DEGREES)	FLOSEP
DELTAE	R	A	CONTROL SURFACE DEFLECTION	FLOSEP
DELTAR	R	A	ELEMENT IMPACT ANGLE (RADIANS)	FLOSEP
DELTAS	R	U	SAVED INITIAL CONTROL SURFACE DEFLECTION	FLOSEP
DELTER	R	U	FLAP DEFLECTION ANGLE (RADIANS)	FLOSEP
DO	R	U	BOUNDARY LAYER THICKNESS	FLOSEP
DOX	R	U	BOUNDARY LAYER THICKNESS AT EXACT SEPARATION POINT	FLOSEP
DO1	R	U	BOUNDARY LAYER THICKNESS ON ELEMENT BEFORE SEPARATION POINT	FLOSEP
DSQXP	R	U	SQUARE ROOT OF X-PRIME	FLOSEP
DUMMY	R	U	DUMMY ARGUMENT	FLOSEP

SYMBOLS USED IN SUBROUTINE FLOSEP

DXSEP	R	U	DIFFERENCE BETWEEN LEADING EDGE AND SEPARATION X-DISTANCE	FLOSEP
DXSEP1	R	U	XLE-XSEP ON ELEMENT JUST BEFORE SEPARATION ELEMENT	FLOSEP
D1	R	U	UPSTREAM INTERACTION LENGTH	FLOSEP
D1DO	R	U	UPSTREAM INTERACTION LENGTH/BOUNDARY LAYER THICKNESS	FLOSEP
D2	R	U	DOWNSTREAM INTERACTION LENGTH TO PEAK PRESSURE	FLOSEP
D2D1	R	U	RATIO OF DOWNSTREAM TO UPSTREAM INTERACTION LENGTHS	FLOSEP
D3	R	U	DOWNSTREAM INTERACTION LENGTH TO PRESSURE RISE	FLOSEP
D3D1	R	U	DOWNSTREAM INTERACTION LENGTH/UPSTREAM INTERACTION LENGTH	FLOSEP
ELAM	R	U	A FUNCTION OF RELATIVE FLAP CHORD LENGTH	FLOSEP
ELFI	R	U	FREE INTERACTION LENGTH	FLOSEP
ELFIDO	R	U	FREE INTERACTION LENGTH / BOUNDARY LAYER THICKNESS	FLOSEP
EMADF	R	U	PRODUCT OF MACH NUMBER AND FLAP DEFLECTION	FLOSEP
ERROR	I	C	ERROR FLAG	FLOSEP
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	FLOSEP
FS2	R	U	PRESSURE TO BE SAVED	FLOSEP
FS6	R	U	MACH NUMBER TO BE SAVED	FLOSEP
G	R	A	SPECIFIC HEAT RATIO (GAMMA)	FLOSEP
I	I	U	DO LOOP COUNTER .	FLOSEP
ICHECK	I	U	PRINT FLAG FOR CHECKOUT PURPOSES	FLOSEP
IFIRST	I	A	FLAG FOR FIRST TIME INTO NEWTONIAN-PRANDTL-MEYER ROUTINE	FLOSEP
IFSCY	I	A	FLOW SEPARATION CYCLE FLAG	FLOSEP
IGT	I	A	CONTROL SURFACE FLAG (=1 FORESURFACE, = 2 CONTROL SURFACE)	FLOSEP
IGTS	I	A	CONTROL SURFACE FLAG FOR THE PRESENT ELEMENT	FLOSEP
IGTYPE	I	A	GEOMETRY TYPE (=1 FOR CONTROL SURFACE COMPONENT)	FLOSEP
IM	I	C	(NOT USED)	FLOSEP
IMPACI	I	A	INITIAL STRIP ELEMENT IMPACT FORCE METHOD	FLOSEP
IMPACT	I	A	IMPACT FORCE CALCULATION METHOD FLAG	FLOSEP
IMPS	I	A	INPUT IMPACT FORCE CALCULATION METHOD	FLOSEP
IN	I	C	IN(1) AND IN(2) NUMBER OF ELEMENTS IN FORE-SURFACE AND FLAP	FLOSEP
IORIEN	I	A	ELEMENT ORIENTATION (=1 FOR STREAMWISE STRIP)	FLOSEP
IPRCK	I	A	DETAILED DATA PRINT CONTROL FLAG	FLOSEP
IRFD11	I	A	TAPE 11 READ FLAG INDICATOR	FLOSEP
ISBP	I	A	FORCE SUMMATION BYPASS FLAG (=1 TO BYPASS SUMMATION)	FLOSEP
ISCT	I	U	ELEMENT COUNTER	FLOSEP
ISDET	I	U	CALCULATION CONTROL FLAG FOR COMPRESSION ROUTINE	FLOSEP
ISEP	I	U	SEPARATION INDICATOR FLAG	FLOSEP
ISHAD	I	A	SHADOW FORCE CALCULATION METHOD FLAG	FLOSEP

SYMBOLS USED IN SUBROUTINE FLOSEP

ISHADI	I	A	INITIAL STRIP ELEMENT SHADOW FORCE METHOD	FLOSEP
ISHS	I	A	INPUT SHADOW FORCE CALCULATION METHOD	FLOSEP
ISPNT	I	A	SEPARATION AND ATTACHMENT PRINT FLAG	FLOSEP
ITRANS	I	U	FLOW TRANSITION FLAG (=0 LAMINAR, =1 TURBULENT)	FLOSEP
J	I	U	DJ LOOP INDEX	FLOSEP
K	I	C	NUMBER OF ELEMENTS IN COMPONENT	FLOSEP
KFLSP	I	U	FLOW SEPARATION FLAG (=0 NO SEPARATION, =1 FLOW SEPARATED)	FLOSEP
L	I	A	NUMBER OF ELEMENT IN FORCE CALCULATION LOOP	FLOSEP
LATT	I	U	ELEMENT NUMBER AT FLOW ATTACHMENT POINT	FLOSEP
LL	I	A	ELEMENT NUMBER	FLOSEP
LS	I	C	NUMBER OF ELEMENTS IN COMPONENT	FLOSEP
LSEP	I	U	ELEMENT NUMBER AT SEPARATION	FLOSEP
LSS	I	U	NUMBER OF ELEMENTS IN COMPONENT	FLOSEP
M	I	A	ELEMENT NUMBER IN STRIP	FLOSEP
MA	R	U	LOCAL MACH NUMBER ON ELEMENT	FLOSEP
MACH	R	A	FREE-STREAM MACH NUMBER	FLOSEP
MACHH	R	U	SHOCK-EXPANSION MACH NUMBER AT HINGE LINE ELEMENT	FLOSEP
MACHX	R	U	MACH NUMBER AT SEPARATION	FLOSEP
MACHX1	R	U	MACH NUMBER ON ELEMENT JUST BEFORE SEPARATION	FLOSEP
MER	I	U	COMPRESSION ROUTINE ERROR FLAG	FLOSEP
N	I	A	STREAMWISE ELEMENT STRIP NUMBER	FLOSEP
NW	R	A	WALL TEMPERATURE CALCULATION FLAG FOR FLOSEP	FLOSEP
NX	R	A	X-COMPONENT OF OUTWARD NORMAL	FLOSEP
NXF	R	U	X-COMPONENT OF FLAP SURFACE NORMAL	FLOSEP
NXFS	R	U	X-COMPONENT OF FLAP OUTWARD NORMAL TO BE SAVED	FLOSEP
NXH	R	U	X-COMPONENT OF OUTWARD NORMAL AT HINGE LINE ELEMENT	FLOSEP
NXS	R	U	X-COMPONENT OF SURFACE NORMAL TO BE SAVED	FLOSEP
NX2	R	C	NX2(1) AND NX2(2) ARE HINGE LINE X-COORDINATE DATA	FLOSEP
NY	R	A	Y-COMPONENT OF OUTWARD NORMAL	FLOSEP
NYF	R	U	Y-COMPONENT OF FLAP OUTWARD NORMAL	FLOSEP
NYFS	R	U	Y-COMPONENT OF FLAP OUTWARD NORMAL TO BE SAVED	FLOSEP
NYH	R	U	Y-COMPONENT OF OUTWARD NORMAL AT HINGE LINE ELEMENT	FLOSEP
NYS	R	U	Y-COMPONENT OF OUTWARD NORMAL TO BE SAVED	FLOSEP
NY2	R	C	NY2(1) AND NY2(2) ARE HINGE LINE Y-COORDINATE DATA	FLOSEP
NZ	R	A	Z-COMPONENT OF OUTWARD NORMAL	FLOSEP
NZF	R	U	Z-COMPONENT OF FLAP SURFACE NORMAL (UNDEFLECTED)	FLOSEP

[redacted]

SYMBOLS USED IN SUBROUTINE FLOSEP

NZFS	R	U	Z-COMPONENT OF FLAP SURFACE NORMAL TO BE SAVED (UNDEFLECTED)	FLOSEP
NZH	R	U	Z-COMPONENT OF OUTWARD NORMAL AT HINGE LINE ELEMENT	FLOSEP
NZS	R	U	Z-COMPONENT OF SURFACE NORMAL TO BE SAVED	FLOSEP
NZ2	R	C	(NOT USED)	FLOSEP
P	R	U	FINAL PRESSURE ON ELEMENT WITH SEPARATION	FLOSEP
PAGE	I	C	PAGE NUMBER	FLOSEP
PFS	R	A	FREE-STREAM PRESSURE	FLOSEP
PH	R	U	HINGE LINE INVISCID SHOCK-EXPANSION PRESSURE	FLOSEP
PHI	R	U	ANGLE ASSOCIATED WITH GEOMETRY OF SEPARATION	FLOSEP
PNIN	R	U	INVISCID PRESSURE USING NORMAL FORCE METHOD	FLOSEP
PPPO	R	U	RATIO OF PLATEAU PRESSURE TO FREE-STREAM PRESSURE	FLOSEP
PPPOX	R	U	PLATEAU PRESSURE/STREAM PRESSURE AT SEPARATION POINT	FLOSEP
PPPO1	R	U	PLATEAU PRESS/FREE-STREAM PRESS ON ELEMENT BEFORE SEPARATION	FLOSEP
PSEIN	R	D	INVISCID SHOCK-EXPANSION PRESSURE	FLOSEP
PSEVIS	R	D	SHOCK-EXPANSION PRESSURE WITH VISCOUS SEPARATION	FLOSEP
PX	R	U	LOCAL PRESSURE AT EXACT SEPARATION POINT	FLOSEP
PX1	R	U	PRESSURE ON ELEMENT JUST BEFORE SEPARATION POINT	FLOSEP
RE	R	D	REFERENCE REYNOLDS NUMBER	FLOSEP
REAFT	R	U	LOCAL SURFACE REYNOLDS NUMBER PER FOOT	FLOSEP
REAHL	R	U	REYNOLDS NUMBER AT HINGE LINE ELEMENT	FLOSEP
REASFT	R	U	REFERENCE REYNOLDS NUMBER PER FOOT	FLOSEP
REAXOP	R	U	REYNOLDS NUMBER ON LOCAL ELEMENT	FLOSEP
RETRAN	R	A	INPUT FLOW TRANSITION REYNOLD'S NUMBER	FLOSEP
RT	R	U	RECOVERY TEMPERATURE	FLOSEP
SINTH	R	U	SINE OF SHOCK ANGLE	FLOSEP
SWEEP	R	A	LEADING EDGE SWEEP ANGLE	FLOSEP
TANPHI	R	U	TANGENT OF FLOW SEPARATION ANGLE	FLOSEP
TFS	R	A	FREE-STREAM TEMPERATURE	FLOSEP
TH	R	U	FLOW TEMPERATURE AT HINGE LINE ELEMENT	FLOSEP
THETA	R	U	SHOCK ANGLE	FLOSEP
TITLE	R	C	TITLE ARRAY	FLOSEP
TR	R	D	TEMPERATURE DATA ARRAY	FLOSEP
TS	R	D	REFERENCE TEMPERATURE (DEGREE R)	FLOSEP
TWALL	R	A	WALL TEMPERATURE FOR FLOSEP	FLOSEP
TX1	R	U	FLOW TEMPERATURE ON ELEMENT JUST BEFORE SEPARATION ELEMENT	FLOSEP
XATACH	R	A	X-COORDINATE AT FLOW ATTACHMENT POINT	FLOSEP

SYMBOLS USED IN SUBROUTINE FLOSEP

XCENT	R	A	HINGE MOMENT FACTOR FOR CONTROL SURFACE ELEMENT	FLOSEP
XCENT2	R	C	HINGE MOMENT FACTOR ARRAY FOR CONTROL SURFACE ELEMENTS	FLOSEP
XHL	R	U	ELEMENT AVERAGE X-COORDINATE AT HINGE LINE	FLOSEP
XLE	R	A	X-DISTANCE FROM CENTROID OF ELEMENT TO LEADING EDGE LINE	FLOSEP
XLEH	R	U	X-DISTANCE FROM LEADING EDGE TO HINGE LINE	FLOSEP
XLESEP	R	U	DISTANCE FROM LEADING EDGE TO SEPARATION POINT	FLOSEP
XLE1	R	U	ELEMENT CENTROID DISTANCE FROM L. E. BEFORE SEPARATION	FLOSEP
XPA	R	A	X-COORDINATES OF QUADRILATERAL ELEMENT	FLOSEP
XSEP	R	U	DISTANCE FROM LEADING EDGE MINUS UPSTREAM INTERACTION LENGTH	FLOSEP
XSEPP	R	A	X-COORDINATE AT FLOW SEPARATION POINT	FLOSEP
XTE	R	U	AVERAGE X-COORDINATE AT TRAILING EDGE	FLOSEP
YCENT	R	A	(NOT USED)	FLOSEP
YCENT2	R	C	(NOT USED)	FLOSEP
YHL	R	U	ELEMENT AVERAGE Y-COORDINATE AT HINGE LINE	FLOSEP
YPA	R	A	Y-COORDINATES OF QUADRILATERAL ELEMENT	FLOSEP
YTE	R	U	AVERAGE Y-COORDINATE AT TRAILING EDGE	FLOSEP
ZCENT	R	A	(NOT USED)	FLOSEP
ZCENT2	R	C	(NOT USED)	FLOSEP
ZHL	R	U	ELEMENT AVERAGE Z-COORDINATE AT HINGE LINE	FLOSEP
ZPA	R	A	Z-COORDINATES OF QUADRILATERAL ELEMENT	FLOSEP
ZTE	R	U	ELEMENT AVERAGE Z-COORDINATE AT TRAILING EDGE	FLOSEP

13. SUBROUTINE SKINFR (DECK AROL)

This routine calculates viscous forces for both laminar and turbulent flows including viscous-interaction and planform effects. Also, the viscous induced pressures are determined.

a. Algorithm

The basic program constants and control flags are first established. The routine then starts a major DO-loop to calculate the viscous forces on each of the skin-friction surfaces. This involves the following steps: Calculate the surface planform geometry, local flow conditions, surface temperature, viscous-interaction effects, laminar and turbulent viscous forces, and summation of forces for either laminar or turbulent as specified. If desired, the induced pressure increment due to viscous-interaction is also calculated. At the user's option, the skin friction may be calculated by the reference temperature, reference enthalpy, or theSpalding-Chi methods.

b. Input/Output

If IS(I, 7) = 1, skin friction data for each surface will be printed.

c. Error

An error condition will occur if the input Mach number is subsonic.

d. Subroutines Required

TEMP, CONE, COMPR, EXPAND, NEWTPM, HEADER

e. Argument List

(ALPHA, CA, SREF, SKIN, NS, ALT, MACH, CPSTAG, TFS, PFS, AFS, RHOFS, IFIRST, VIS, ISIZ, CN, IGTYPE, DELTA)

f. Length

10,030 bytes

DECK AROL

```
      SUBROUTINE SKINFR (ALPHA,CA,SREF,SKIN,NS,ALT,MACH,CPSTAG,TFS,
     1  PFS,AFS,RHOFS,IFIRST,VIS,ISIZ,CN,IGTYPE,DELTA)
C
C
C
C   THIS SUBROUTINE CALCULATES SKIN FRICTION FOR BOTH LAMINAR
C   AND TURBULENT FLOWS.  REFERENCE TEMPERATURE, REFERENCE
C   ENTHALPY, AND SPALDING-CHI CALCULATION PROCEDURES MAY BE SELECTED.
C   ALSO, THE INDUCED PRESSURES DUE TO VISCOUS-INTERACTION ARE
C   DETERMINED IN THIS SUBROUTINE.
C
C
C
      DIMENSION  ANGLE(3),BS(8),CFT(6),CFL(6),FS(8),RE(2),SCF(4),TR(10),
     1  TS(2),          SCFA(2),TITLE(15),SURF(10,8),IS(10,9),RT(2),
     2  CCA(20),CCY(20),CCN(20),CCLL(20),CCLM(20),CCLN(20),CCL(20),
     3  CCD(20),CLOD(20),CF(20),CPS(20),ETACS(20),ALP(20),BET(20)
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),
     1  YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)
C
C
      COMMON  CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1  AREA2,IN,IM,L,LS,FS,BS,ALP,BET,CCA,CCY,CCN,CCLL,CCLM,CCLN,CCL,
     2  CCD,CLOD,CF,CPS,ETACS,IS,SURF,NPRT
      COMMON /TEMPOC/HAW,H2,H1,HW,CKU,FC,FRX,RET,ELLOC,CCP,  YST1,ROMURA
      INTEGER  CASE,PAGE,ERROR
      REAL  MACH, NX2, NY2, NZ2, KO, M, LAMBDA
C
C
C   CHECK IF FLOW IS SUPERSONIC
    2  IF (MACH .GT.1.0) GO TO 7
      WRITE (6,9)
    9  FORMAT (1H0,47H**** INPUT MACH NUMBER IS NOT SUPERSONIC.  SKIN
     1 49H FRICTION ANALYSIS FOR THIS POINT IS STOPPED ****  )
      SKIN = 0.0
```

DECK AROL

```
      GO TO 1                                                            AROL   0360
C                                                                        AROL   0370
C   SET STARTING CONSTANTS                                               AROL   0380
    7 MER   = 1                                                          AROL   0390
      SCFA(1) = 0.0                                                      AROL   0400
      SCFA(2) = 0.0                                                      AROL   0410
      IF (IGTYPE.NE.2 .OR. (IGTYPE.EQ.2 .AND. NS.EQ.1)) NPRY = 0         AROL   0420
      IGTY = 0                                                           AROL   0430
      IF (IGTYPE .EQ. 17) IGTY = 1                                       AROL   0440
      IF (IGTYPE .EQ. 17) IGTYPE = 2                                     AROL   0450
      MEREXP= 0                                                          AROL   0460
      NC    = 0                                                          AROL   0470
      SCF(1) = 0.0                                                       AROL   0480
      SCF(2) = 0.0                                                       AROL   0490
      SCF(3) = 0.0                                                       AROL   0500
      SCF(4) = 0.0                                                       AROL   0510
      FS(1) = RHOFS                                                      AROL   0520
      FS(2) = PFS                                                        AROL   0530
      FS(3) = TFS                                                        AROL   0540
      FS(4) = AFS                                                        AROL   0550
C                                                                        AROL   0560
C   VISCOSITY EQUATION                                                   AROL   0570
      FS(5) = VIS                                                        AROL   0580
C                                                                        AROL   0590
      TR(1) = ALT                                                        AROL   0600
      TR(2) = MACH                                                       AROL   0610
      TR(3) = MACH * FS(4)                                               AROL   0620
      TR(4) = ALPHA + 0.000001                                           AROL   0630
      IF (IGTYPE .EQ. 2) TR(4) = 0.0                                     AROL   0640
      FS(6) = TR(2)                                                      AROL   0650
      FS(7) = TR(3)                                                      AROL   0660
      FS(8) = FS(1)*FS(7)/FS(5)                                          AROL   0670
      II = 1                                                             AROL   0680
      IF (IGTYPE .EQ. 2) II = NS                                         AROL   0690
C                                                                        AROL   0700
C                                                                        AROL   0710
```

```
DECK AROL

C  CALCULATE SKIN FRICTION FOR EACH SURFACE                                  AROL  0720
      DO 1000 I=II,NS                                                         AROL  0730
      ELO = 0.0                                                               AROL  0740
      EL  = SURF(I,2)                                                         AROL  0750
C                                                                             AROL  0760
C  CHECK IF INITIAL SURFACE SPECIFIED (IGTYPE = 2).  IF SO,                   AROL  0770
C  DETERMINE APPROPRIATE LENGTHS AND TAPER RATIOS.                            AROL  0780
      IF (IGTYPE.NE.2) GO TO 110                                              AROL  0790
      ELO = SURF(I,3)                                                         AROL  0800
      TAPER1 = SURF(I,4)                                                      AROL  0810
      TAPER2 = SURF(I,5)                                                      AROL  0820
      IF (TAPER1 .LT. 0.0) GO TO 130                                          AROL  0830
      SURF(I,5) = (EL*TAPER2 + ELO*TAPER1)/(EL + ELO)                         AROL  0840
      EL = EL + ELO                                                           AROL  0850
      GO TO 133                                                               AROL  0860
  130 TAPER1 = -TAPER1                                                        AROL  0870
      SURF(I,5) = 1.0                                                         AROL  0880
      EL1 = ELO + EL*TAPER2                                                   AROL  0890
      EL = EL + ELO*TAPER1                                                    AROL  0900
      IF (EL-EL1) 131,133,132                                                 AROL  0910
  131 SURF(I,5) = EL/EL1                                                      AROL  0920
      EL = EL1                                                                AROL  0930
      GO TO 133                                                               AROL  0940
  132 SURF(I,5) = EL1/EL                                                      AROL  0950
  133 IF (TAPER1 .EQ. 0.0) TAPER1 = 0.0001                                    AROL  0960
  110 IF (SURF(I,5) .EQ. 0.0) SURF(I,5) = 0.0001                              AROL  0970
      TR(5) = SURF(I,6)                                                       AROL  0980
       TR(6) = SURF(I,7)                                                      AROL  0990
C                                                                             AROL  1000
      IF (SURF(I,5) .LT. 0.8) GO TO 77                                        AROL  1010
      ELL     = EL * 4.0 * ((1.0 + SURF(I,5)) / (3.0+SURF(I,5)))**2           AROL  1020
      GO TO 79                                                                AROL  1030
C                                                                             AROL  1040
   77 ELL     = EL * (0.75*(1.0-SURF(I,5)**2) /(1.0-SURF(I,5)**1.5))**2       AROL  1050
   79 RENO = FS(8) * ELL                                                      AROL  1060
      IS3 = IS(I,3)                                                           AROL  1070
```

```
      DECK AROL
C                                                                       AROL
C  DETERMINE ANGLE OF ATTACK USE                                        AROL
      IF (IS(I,5) .NE. 1) GO TO 14                                      AROL
C                                                                       AROL
C  ANGLE OF ATTACK IS NOT USED                                          AROL
      ANGLE(1) = SURF(I,4)                                              AROL
      GO TO 90                                                          AROL
C                                                                       AROL
C  ANGLE OF ATTACK IS USED. CHECK CALCULATION TYPE REQUIRED             AROL
   14 GO TO (21,16,21,16,21,16), IS3                                    AROL
C                                                                       AROL
C  UPPER SURFACE                                                        AROL
   21 ANGLE(1) = SURF(I,3) + TR(4) - SURF(I,4)                          AROL
      IF (IGTYPE .EQ. 2) ANGLE(1) = -DELTA                              AROL
C  CHECK FLOW TYPE                                                      AROL
      IF (ANGLE(1)) 18,19,17                                            AROL
C                                                                       AROL
C  LOWER SURFACE                                                        AROL
   16 ANGLE(1) = SURF(I,3) + TR(4) + SURF(I,4,                          AROL
      IF (IGTYPE .EQ. 2) ANGLE(1) = DELTA                               AROL
C  CHECK FLOW TYPE                                                      AROL
   90 IF (ANGLE(1)) 17,19,18                                            AROL
C                                                                       AROL
C                                                                       AROL
C  COMPRESSION                                                          AROL
   18 KF = 2                                                            AROL
C  FLAT PLATE OR DELTA WING CHECK                                       AROL
      IF (IS(I,2) .NE. 1) GO TO 86                                      AROL
      CALL CONE (ANGLE,CP,0)                                            AROL
      GO TO 24                                                          AROL
C                                                                       AROL
   86 ANGLE(2) = ABS(ANGLE(1))                                          AROL
C  CHECK USE OF SHOCK-EXPANSION                                         AROL
      IF (IS(I,3) .LT. 5) GO TO 23                                      AROL
       KF = 5                                                           AROL
       GO TO 85                                                         AROL
```

```
DECK AROL

C                                                                      AROL  1440
   23  ISE = 0                                                         AROL  1450
       ISDET = 0                                                       AROL  1460
       CALL COMPR (ANGLE,MER,IS(I,9),CPSTAG,TFS,PFS,ISDET,IFIRST,CP)   AROL  1470
        GO TO 67                                                       AROL  1480
C                                                                      AROL  1490
C                                                                      AROL  1500
C  EXPANSION                                                           AROL  1510
   17  ANGLE(2) = ABS(ANGLE(1))                                        AROL  1520
C  CHECK USE OF SHOCK-EXPANSION                                        AROL  1530
       IF (IS(I,3) .LT. 3) GO TO 22                                    AROL  1540
        KF = 4                                                         AROL  1550
        ANGLE(2) = -ANGLE(2)                                           AROL  1560
        GO TO 85                                                       AROL  1570
C                                                                      AROL  1580
   22  KF = 1                                                          AROL  1590
       ISDET = 0                                                       AROL  1600
       CALL EXPAND (ANGLE,MER,IS(I,9),ISDET,CP)                        AROL  1610
       IF (IGTY.EQ.0 .OR. MER.LE.1) GO TO 67                           AROL  1620
       SKIN = 0.0                                                      AROL  1630
       GO TO 1                                                         AROL  1640
C                                                                      AROL  1650
C                                                                      AROL  1660
C  NEITHER EXPANSION OR COMPRESSION                                    AROL  1670
   19  ANGLE(2) = 0.0                                                  AROL  1680
C  CHECK USE OF SHOCK-EXPANSION                                        AROL  1690
       IF (IS(I,3).LT. 5) GO TO 84                                     AROL  1700
        KF = 6                                                         AROL  1710
        GO TO 85                                                       AROL  1720
C                                                                      AROL  1730
   84  KF = 3                                                          AROL  1740
       DO 20 K=1,8                                                     AROL  1750
   20   BS(K) = FS(K)                                                  AROL  1760
        ONEOM = 1.0 / BS(6)                                            AROL  1770
       ANGLE(3) = ARSIN(ONEOM) * 0.5729578E02                          AROL  1780
        GO TO 24                                                       AROL  1790
```

```
DECK AROL

C                                                                      AROL
C                                                                      AROL
C  SHOCK-EXPANSION CALCULATION                                         AROL
  85  ISE = 0                                                          AROL
      CALL NEWTPM (ANGLE,EMN,CP,SURF(I,8),IS(I,9),MER,CPSTAG,          AROL
     1  TFS, PFS,ISE,IFIRST)                                           AROL
      IF (MER .EQ. 0) GO TO 67                                         AROL
      ISDET = 1                                                        AROL
      CALL COMPR (ANGLE,MER,IPRINT,CPSTAG,TFS,PFS,ISDET,IFIRST,CP)     AROL
C                                                                      AROL
C  CHECK ERROR FLAG                                                    AROL
  67  IF (MER .LE. 1) GO TO 24                                         AROL
       MEREXP = 1                                                      AROL
   5   MER = 1                                                         AROL
       GO TO 65                                                        AROL
C                                                                      AROL
C  CALCULATE REFERENCE TEMPERATURE, REYNOLDS NO., AND WALL             AROL
C  TEMPERATURE (IF REQUIRED).  FIRST CHECK IF TEMPERATURE              AROL
C  ITERATIONS AND/OR LOCAL CF DATA TO BE PRINTED.                      AROL
   24 IF (IS(I,9).EQ.0) GO TO 26                                       AROL
C                                                                      AROL
C  LOCAL DATA TO BE PRINTED.  IF TEMPERATURE ITERATIONS START          AROL
C  NEW PAGE FOR EACH SURFACE.  OTERWISE TEST IF HEADER REQUIRED.       AROL
      IF (IS(I,9).EQ.1) GO TO 25                                       AROL
      IF ((NPRT.LT.52).AND.(NS.NE.1))GO TO 26                          AROL
C                                                                      AROL
   25 CALL HEADER                                                      AROL
      WRITE(6,31) ALPHA,MACH,TR(3),ALT,FS(8),SREF                      AROL
      NPRT = 8                                                         AROL
   26 CALL TEMP(EL,TR,RE,TS,IS(I,6),MER,IS(I,9),RT)                    AROL
C  CHECK ERROR FLAG                                                    AROL
      IF (MER - 1) 83,83,5                                             AROL
C                                                                      AROL
C  CALCULATE VISCOUS-INVISCID INTERACTION EFFECT ON SKIN FRICTION      AROL
  83  KO = MACH * SIN(ANGLE(1) / 57.295779)                            AROL
      IF (IS3.EQ.1.OR.IS3.EQ.3.OR.IS3.EQ.5) KO = - KO                  AROL
```

DECK AROL

```
      IF (KO.LT.-0.0001 .OR. KO.GT.0.0001) GO TO 91
      PO = 1.0
       M = 1.9156
       GO TO 92
   91 GA = 1.4
      IF (KO.GE.0.0) GO TO 93
C  EXPANSION SURFACE.
      PO = 0.00001
      M = 0.00001
      IF (KO.LE.(-2/(GA-1.))) GO TO 92
      PO = (1.0 + 0.5*(GA-1.)*KO)**(2.*GA/(GA-1.))
      IF (KO.GE.-3.0) GO TO 94
      M = 1.424 + 0.219*KO
      GO TO 92
C  COMPRESSION SURFACE.
   93 PO = 1.0 + 0.25*GA*(GA + 1.0)*KO*KO +
     1         GA*KO*SQRT(1.0 + (0.25*(GA + 1.0)*KO)**2)
   94 M = 1.9156 + KO*(0.41727 - KO*(0.0419101 + KO*(0.010427
     1          - KO*(0.00214381 - KO*1.03217E-4))))
  92  TWTRL  = TR(5) / RT(1)
      G = 0.34416 * (TWTRL + 0.3859)
      A =  G / 2.0
      IF (TR(5).GE.225.0) VISWL =2.27*TR(5)**1.5/((TR(5)+198.6)*10.**8)
      IF (TR(6).GE.225.0) VISWT =2.27*TR(6)**1.5/((TR(6)+198.6)*10.**8)
      IF (TR(5).LT.225.0) VISWL = 0.80382436E-9 * TR(5)
      IF (TR(6).LT.225.0) VISWT = 0.80382436E-9 * TR(6)
      RELOC = BS(8) * SURF(I,2)
      CLAM = VISWL * TFS / (VIS*TR(5))
      CTURB= VISWT * TFS / (VIS*TR(6))
      CHIBAR = MACH**3*SQRT(CLAM) / SQRT(RENO*EL/ELL)
      VBAR = CHIBAR /(MACH*MACH)
      IF (TS(1).GE.225.0) VISTAR =2.27*TS(1)**1.5/((TS(1)+198.6)*10.**8)
      IF (TS(1).LT.225.0) VISTAR = 0.803824E-9 * TS(1)
      CSTAR = VISTAR * TFS / (VIS*TS(1))
      VSTAR = MACH * SQRT(CSTAR) / SQRT(RENO*EL/ELL)
      FJ = IS(I,2)
```

DECK AROL

```
      LAMBDA = A * CHIBAR /SQRT(1. + 2.*FJ)
      IF (IGTY .EQ. 1) GO TO 500
      B = M * LAMBDA / PO * SQRT(EL/ELL)
      CFCFOL = SQRT(1.0+B) + 0.5*B*ALOG(ABS((SQRT(1.0+B)+1.0)/
     1                (SQRT(1.0+B)-1.0)))
      IF (CFCFOL .LT. 1.0) CFCFOL = 1.0
      CFCFOT = 1.0

C
C  START TURBULENT FLOW CALCULATIONS
      IF (RE(2) .LE. 6570.) GO TO 100
      EN = 0.8686 / (ALOG10(RE(2)) - 2.0)
C  CHECK TAPER RATIO AND CHARACTERISTIC LENGTH TERMS
      IF (SURF(I,5) .LT. 0.8) GO TO 72
      Q = SQRT ((1.0 + SURF(I,5)) / (1.0 + EN + SURF(I,5) * (1.0-EN)))
      GO TO 73
C
   72 Q = SQRT ((1.0 - SURF(I,5)**2) *(1.0- 0.5 *EN) /
     1   (1.0 - SURF(I,5)**(2.0-EN)))
C
   73 ELT = EL * (RE(2)/10.0**1.5) ** (Q - 1.0)
      ESIN = SIN((ANGLE(1)-TR(4))/57.295779)
      ECOS = COS((ANGLE(1)-TR(4))/57.295779)
      RE(2)  = RE(2) * ELT/EL
      CFT(1) = 0.088 / (0.43429448 * ALOG(RE(2)) - 1.5)**2
      FF = ELO      /SURF(I,2)*(1.0 + TAPER1)/(1.0 + TAPER2)
      IF (ELO.LT. 0.0001) GO TO 112
      IF ((RE(2)*ELO/ELT).GT. 6570.0) GO TO 111
      IF (TAPER1.LT. 0.8) GO TO 113
      EL1 = ELO*4.0*((1.+TAPER1)/(3.0+TAPER1))**2
      GO TO  114
  113 EL1 = ELO*(0.75*(1.-TAPER1**2)/(1.-TAPER1**1.5))**2
  114 CFL(1) = 1.328/SQRT(RE(2)*EL1/ELT)
      FF = FF*(CFL(1)/CFT(1) - 1.0)
      GO TO 112
  111 EN = 0.8686/(ALOG10(RE(2)*ELO/ELT) - 2.0)
```

DECK AROL

```
      IF (TAPER1.LT. 0.8) GO TO 115
      Q = SQRT((1.0+TAPER1)/(1.0+EN+TAPER1*(1.-EN)))
      GO TO 116
  115 Q = SQRT((1.0-TAPER1**2)*(1.0-0.5*EN)/(1.0-TAPER1**(2.0-EN)))
  116 EL1 = ELO*(RE(2)*ELO/ELT/10.**1.5)**(Q-1.0)
      FF = FF*(((ALOG10(RE(2))-1.5)/(ALOG10(RE(2)*EL1/ELT)
     1              -1.5))**2 - 1.0)
  112 CFT(1) = CFT(1)*(1.0-FF)
      CFT(2) = CFT(1) /FC
      CFT(3) = CFT(2) * (BS(7)/FS(7))**2 * BS(1) / FS(1) * CFCFOT
      CFT(4) = CFT(3) * SURF(I,1) / SREF
      IF (IGTYPE .EQ. 2) GO TO 100
      IF (IS(I,5).EQ.0) CFT(5) = -ESIN * CFT(4)
      IF (IS(I,5).EQ.0) CFT(6) =  ECOS * CFT(4)
      IF (IS(I,5).EQ.1) CFT(5) =  0.0
      IF (IS(I,5).EQ.1) CFT(6) = CFT(4) * COS(SURF(I,4)/57.295779)
C
C  START LAMINAR FLOW CALCULATIONS
 100  RE(1)  = RE(1) * ELL/EL
      ESIN = SIN((ANGLE(1)-TR(4))/57.295779)
      ECOS = COS((ANGLE(1)-TR(4))/57.295779)
      FF = ELO      /SURF(I,2)*(1.0 + TAPER1)/(1.0 + TAPER2)
      IF (ELO.LT. 0.0001) GO TO 122
      IF (TAPER1.LT. 0.8) GO TO 120
      EL1 = ELO*4.0*((1.0 + TAPER1)/(3.0 + TAPER1))**2
      GO TO 121
  120 EL1 = ELO*(0.75*(1.0-TAPER1**2)/(1.0-TAPER1**1.5))**2
  121 B = B*SQRT(ELL/EL1)
      CFCFO1 = SQRT(1.0+B) + 0.5*B*ALOG(ABS((SQRT(1.0+B)+1.0)/
     1              (SQRT(1.0+B)-1.0)))
      FF = FF*(SQRT(ELL/EL1)*CFCFO1/CFCFOL - 1.0)
  122 CFL(1) = 1.328/SQRT(RE(1))*(1.0-FF)
      CFL(2) = CFL(1) * BS(3)/TS(1)
      CFL(3) = CFL(2) * (BS(7)/FS(7))**2 * BS(1) / FS(1) * CFCFOL
      CFL(4) = CFL(3) * SURF(I,1) / SREF
      IF (IGTYPE .EQ. 2) GO TO 101
```

DECK AROL

```
      IF (IS(I,5).EQ.0) CFL(5) = -ESIN * CFL(4)
      IF (IS(I,5).EQ.0) CFL(6) =  ECOS * CFL(4)
      IF (IS(I,5).EQ.1) CFL(5) =  0.0
      IF (IS(I,5).EQ.1) CFL(6) = CFL(4) * COS(SURF(I,4)/57.295779)
C
C  CALCULATE TOTALS USING LAMINAR FLOW
      SCF(1) = SCF(1) + CFL(5)
      SCF(2) = SCF(2) + CFL(6)
C  TOTAL SKIN FRICTION FORCE COEFF. AND SURFACE
      SCFA(1) = SCFA(1) + CFL(4)
C
C  CHECK IF RE(2) IS LOWER THAN THE CUTOFF POINT (USE LAMINAR PROP.)
      IF (RE(2) .GT. 6570.0) GO TO 82
      SCF(3) = SCF(3) + CFL(5)
      SCF(4) = SCF(4) + CFL(6)
      SCFA(2) = SCFA(2) + CFL(4)
      CFT(3) = CFL(3)
      CFT(5) = CFL(5)
      CFT(6) = CFL(6)
      GO TO 28
C  CALCULATE TOTALS USING TURBULENT FLOW
   82 SCF(3) = SCF(3) + CFT(5)
      SCF(4) = SCF(4) + CFT(6)
C  TOTAL SKIN FRICTION FORCE COEFF. AND SURFACE
      SCFA(2) = SCFA(2) + CFT(4)
      GO TO 28
  101 CFL(6) = 0.0
      CFL(5) = 0.0
      SCF(2) = 0.0
      SCF(1) = 0.0
      CDL = 0.0
      CFT(6) = 0.0
      CFT(5) = 0.0
      SCF(4) = 0.0
      SCF(3) = 0.0
      CDT = 0.0
```

DECK AROL

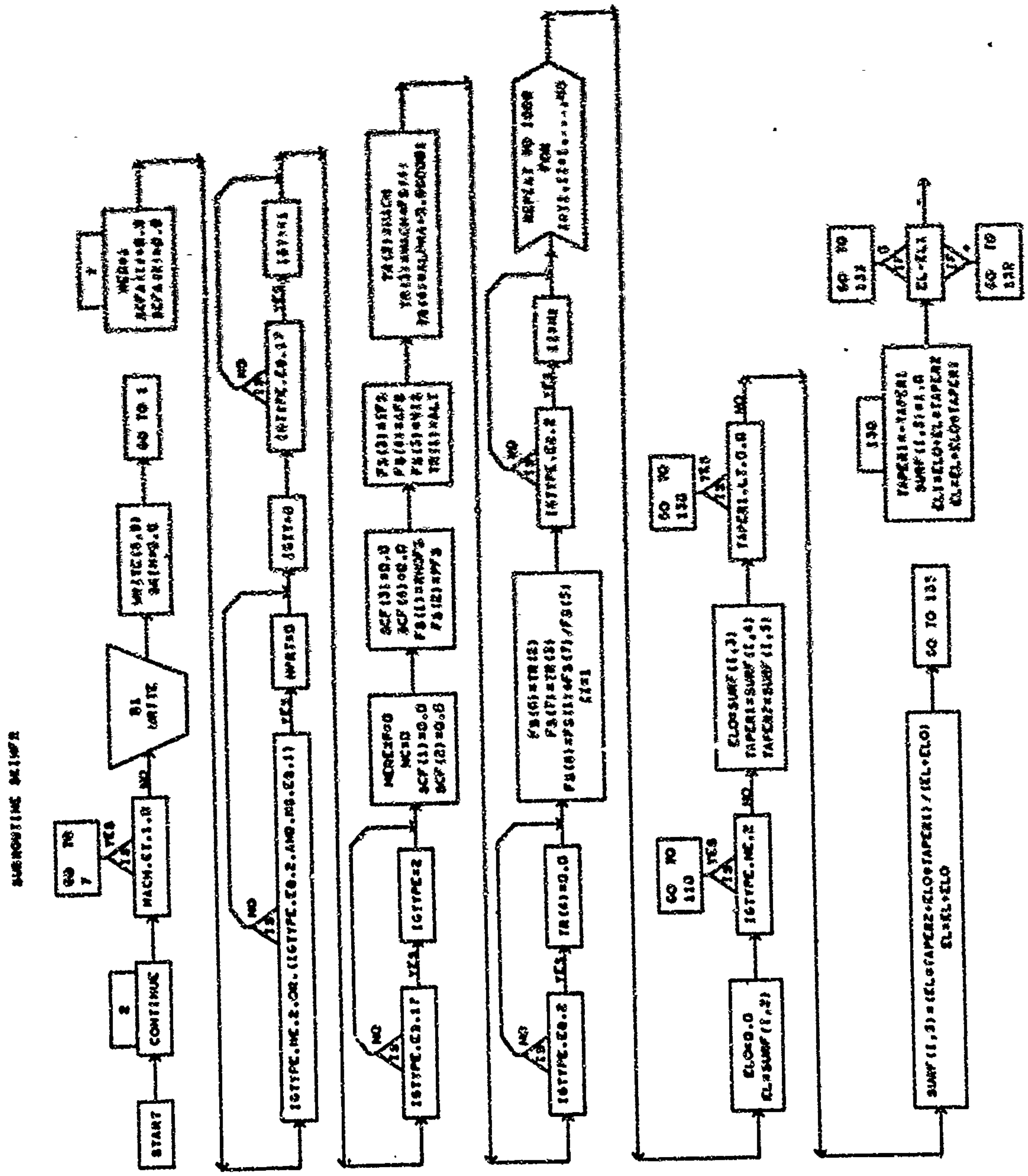
```
      IF (RE(2) .LE. 6570.0) CFT(3) = CFL(3)
C
C  CALCULATE RE*/FT
   28 RE1 = RE(1) / ELL
      IF (RE(2) .LE. 6570.0) ELT = EL
      RE2 = RE(2) / ELT
      TWTL = TR(5) / FS(3)
      TWTT = TR(6) / FS(3)
      TWTRL = TR(5) / RT(1)
      TWTRT = TR(6) / RT(2)
      IF (IGTYPE .EQ. 2) GO TO 65
      ALPHAR = ALPHA / 57.295779
      CDT = CFT(5)*SIN(ALPHAR) + CFT(6)*COS(ALPHAR)
      CDL = CFL(5)*SIN(ALPHAR) + CFL(6)*COS(ALPHAR)
C
C  CHECK IF DATA IS TO BE PRINTED
   65 IF (IS(I,7 ) .EQ. 0) GO TO 1000
C
C  PRINT SKIN FRICTION DATA
C
      IF (IS(I,9).EQ.0) GO TO 32
      IF (IS(I,9).EQ.1) GO TO 33
      IF ((NS.NE.1).AND.(NPRT.GT.23)) GO TO 60
      GO TO 33
C
C  CHECK IF HEADER IS REQUIRED
   32 IF ((NS.NE.1).AND.(NPRT.LT.52)) GO TO 60
      CALL HEADER
      NPRT = 4
C
      WRITE (6,31) ALPHA,MACH,TR(3),ALT,FS(8),SREF
   31 FORMAT (1H0,
     1 23H FREE STREAM CONDITIONS,/1H ,9X7HALPHA =F7.2,6X9HMACH    =
     2 F6.2,3X10HVELOCITY =F9.1,3X10HALTITUDE =F9.1,/1H ,9X7HRE/FT =
     3 1PE10.3,3X9HS REF    =0PF9.1)
      NPRT = NPRT + 4
```

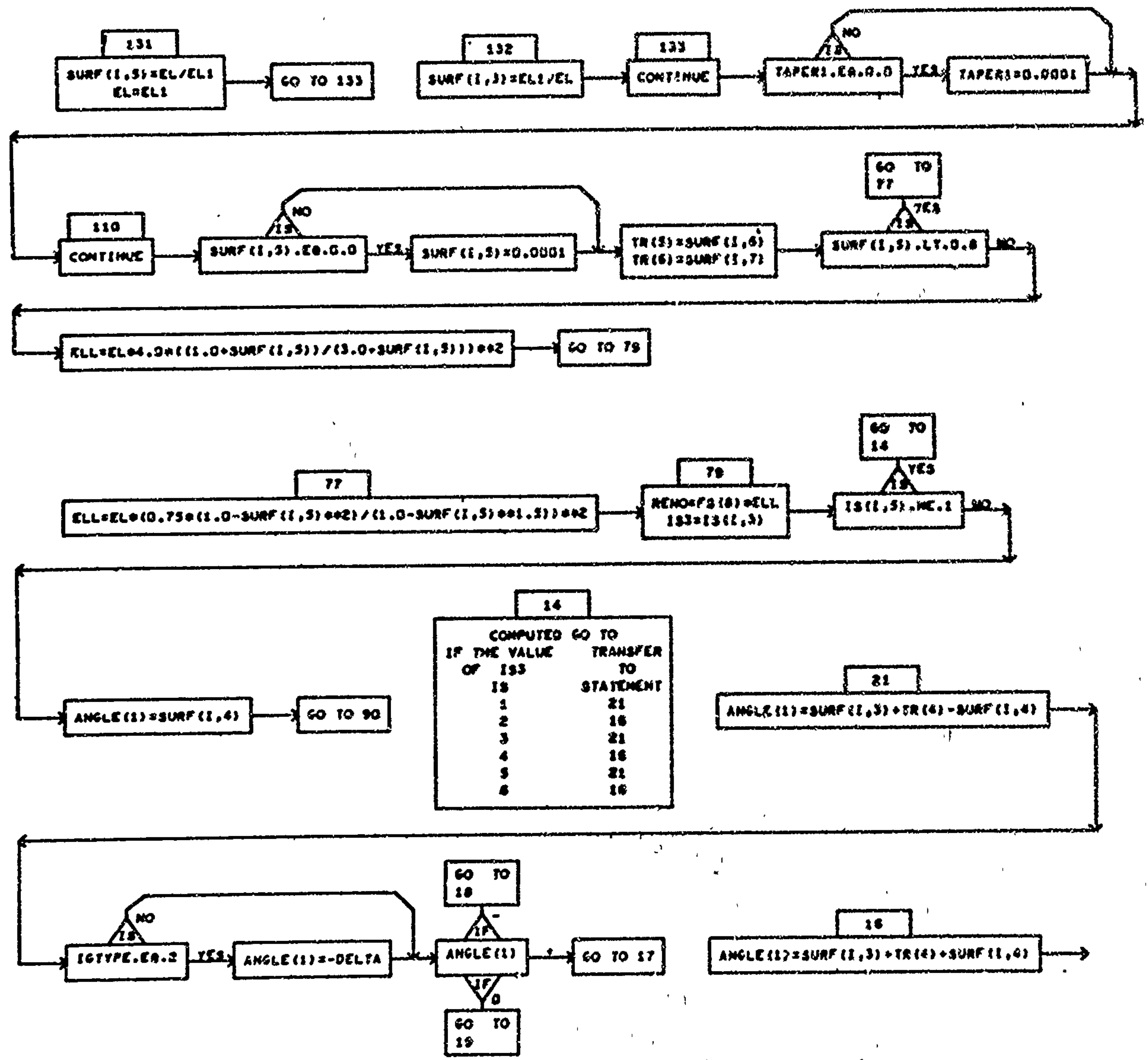
```
DECK AROL
C
   33 WRITE(6,58)
      NPRT = NPRT + 5
   58 FORMAT (1H0,19H SKIN FRICTION DATA,/1H ,9X8HSURF NO.,3X4HTYPE,
     1 4X6HMETHOD,4X5HS WET,3X6HLENGTH,3X7HALPHA O,3X5HWEDGE,
     2 4X8HANGLE(2),2X6HRE LOC,4X7HCHI BAR,2X5HV BAR,/
     3 1H ,2X3HLAM,7X2HCF,7X2HCA,7X2HCN,5X6HSUM CA,3X6HSUM CN,5X2HTW,
     4 7X4HTW/T,6X5HTW/TR,3X6HRE*/FT,4X7H   CD   ,2X6HCF/CFO,/
     5 1H ,2X4HTURB,6X2HCF,7X2HCA,7X2HCN,5X6HSUM CA,3X6HSUM CN,5X2HTW,
     6 7X4HTW/T,6X5HTW/TR,3X6HRE*/FT,4X7H   CD   ,2X6HCF/CFO )
C
      IF (IS(I,8) .EQ. 0) GO TO 60
      WRITE (6,4)
      NPRT = NPRT + 1
    4 FORMAT (1H ,10X4HMACH,8X1HV,5X7HV SCUND,3X5HP-PSF,3X6HTEMP-R,
     1 3X9HRHO*10**4,1X9HVIS*10**7,3X5HRE/FT,2X,6HC STAR,7X,1HC,5X,
     2 6HV STAR)
C
   60 WRITE (6,39) (IS(I,J),J=1,3),(SURF(I,J),J=1,4),ANGLE(2),RELOC
     1             ,CHIBAR,VBAR
      NPRT = NPRT + 2
   39 FORMAT (1H0,11XI2,9XI1,7XI2,4XF8.0,2XF6.1,4XF5.1,3XF5.1,6XF6.2,
     1 1X1PE10.3,2X0PF7.3,1XF7.4 )
C
      IF (MEREXP.NE.0) GO TO 59
C
      WRITE (6,37)  CFL(3),CFL(6),CFL(5),SCF(2),SCF(1),TR(5),TWTL,
     1  TWTRL,RE1,CDL      ,CFCFOL,
     2              CFT(3),CFT(6),CFT(5),SCF(4),SCF(3),TR(6),TWTT,
     3  TWTRT,RE2,CDT      ,CFCFOT
      NPRT = NPRT + 2
   37 FORMAT (1H ,2X3HLAM,4XF8.5,F9.5,F9.5,F9.5,F9.5,F9.1,F10.4,
     1 F10.4,1PE10.3,0PF9.5,1XF7.4,/
     2 1H ,2X4HTURB,3XF8.5,F9.5,F9.5,F9.5,F9.5,F9.1,F10.4,F10.4,
     3 1PE10.3,0PF9.5,1XF7.4 )
C
```

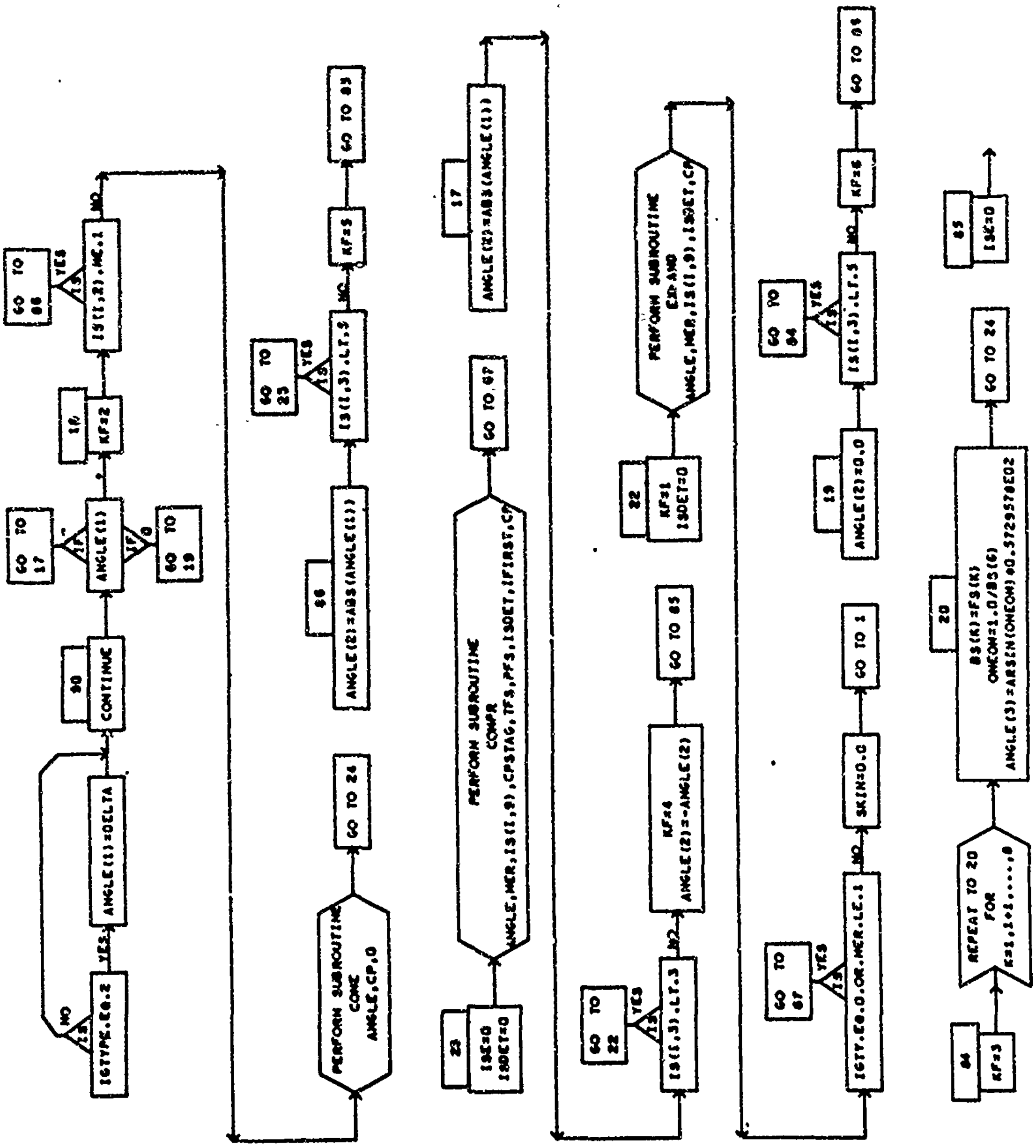
DECK AROL

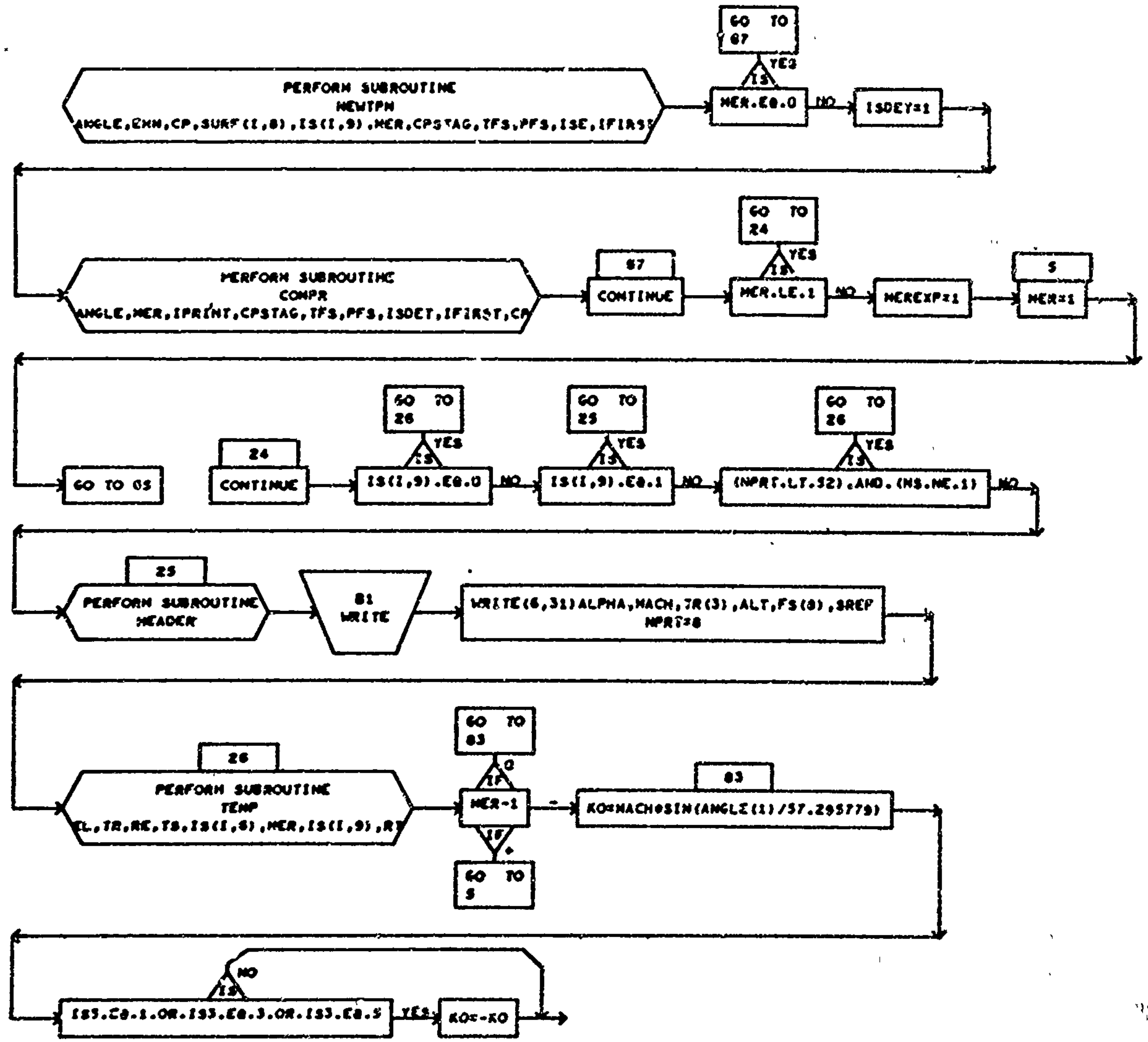
```
   66  IF (IS(I,8) .EQ. 0) GO TO 63                                          AROL  4320
       WRITE (6,44) FS(6),FS(7),FS(4),FS(2),FS(3),FS(1),FS(5),FS(8),         AROL  4330
      1 CSTAR,CLAM,VSTAR,BS(6),BS(7),BS(4),BS(2),BS(3),BS(1),BS(5),BS(8)     AROL  4340
       NPRT = NPRT + 2                                                       AROL  4350
   44  FORMAT (1H ,2X6HSTREAM,F9.5,F9.1,F9.2,F9.4,F9.2,4PF10.7,7PF10.6,      AROL  4360
      11PE10.3,2E10.3,0PF7.4,/1H ,2X6HLOCAL ,0PF9.5,F9.1,F9.2,F9.4,          AROL  4370
      2  F9.2,4PF10.7,7PF10.6,1PE10.3)                                       AROL  4380
       GO TO 63                                                              AROL  4390
C                                                                            AROL  4400
   59  WRITE (6,64)                                                          AROL  4410
       NPRT = NPRT + 1                                                       AROL  4420
   64  FORMAT (1H ,10X43HFLOW EXPANDED TO A VACUUM - SURFACE DELETED )       AROL  4430
       MEREXP = 0                                                            AROL  4440
       CFL(3) = 0.0                                                          AROL  4450
       CFT(3) = 0.0                                                          AROL  4460
       GO TO 66                                                              AROL  4470
C  STEP PRINT FLAG                                                           AROL  4480
   63 CONTINUE                                                               AROL  4490
C                                                                            AROL  4500
C  END OF DO LOOP CALCULATING SKIN FRICTION FOR EACH SURFACE                 AROL  4510
 1000 IJ = I                                                                 AROL  4520
C                                                                            AROL  4530
C                                                                            AROL  4540
C  ADD SKIN FRICTION DRAG TO AXIAL FORCE COEFFICIENT                         AROL  4550
       IF (IGTYPE .EQ. 2) GO TO 102                                          AROL  4560
       IF (IS(I,4) .EQ. 0) GO TO 27                                          AROL  4570
       CA = CA + SCF(2)                                                      AROL  4580
       CN = CN + SCF(1)                                                      AROL  4590
       SKIN = SCF(2)                                                         AROL  4600
       GO TO 1                                                               AROL  4610
   27  CA = CA + SCF(4)                                                      AROL  4620
       CN = CN + SCF(3)                                                      AROL  4630
       SKIN = SCF(4)                                                         AROL  4640
       GO TO 1                                                               AROL  4650
  102  IF (IS(IJ,4) .EQ. 0) SKIN = CFT(3)                                    AROL  4660
       IF (IS(IJ,4) .NE. 0) SKIN = CFL(3)                                    AROL  4670
```

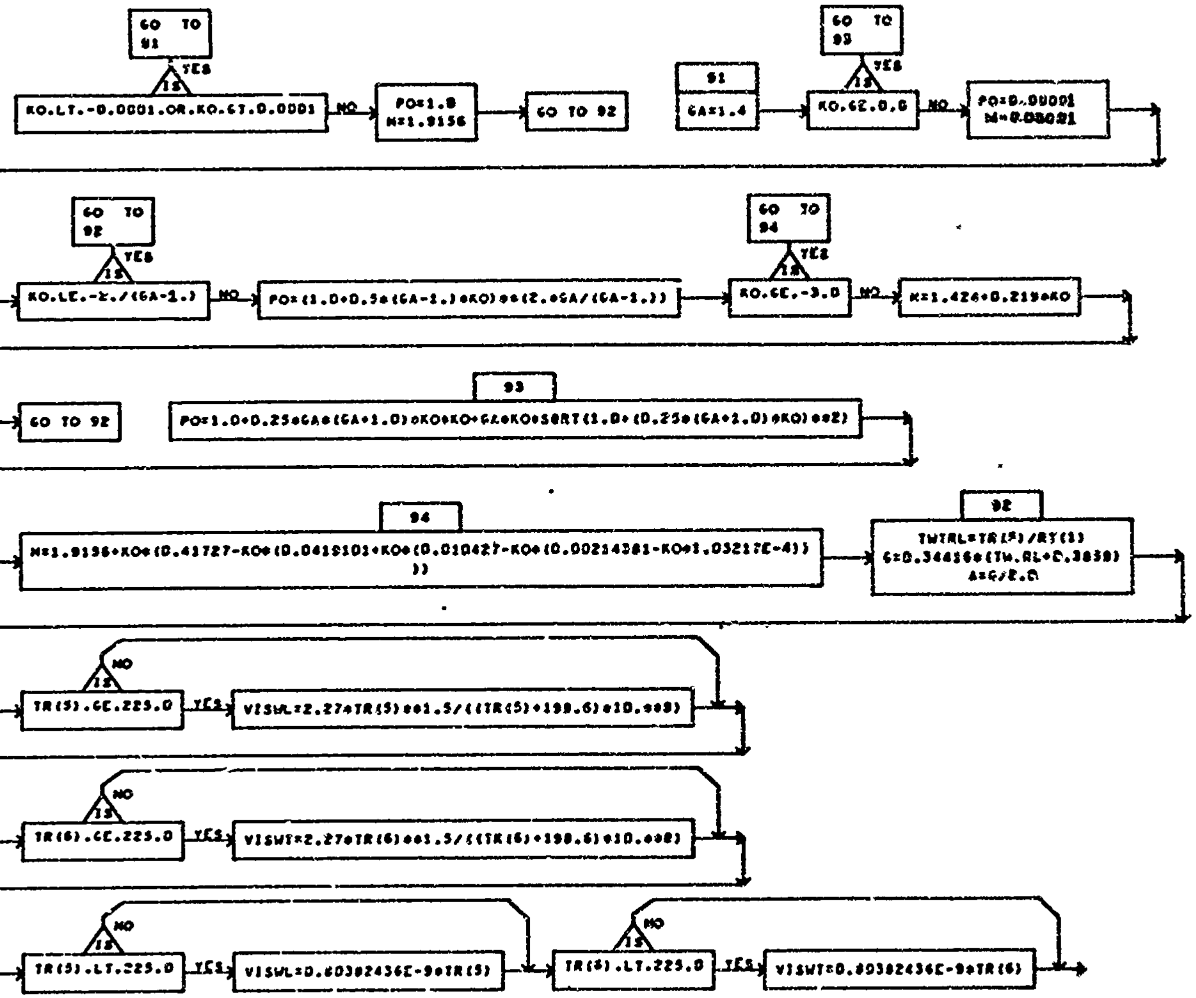
```
DECK AROL

      GO TO 1
C
C  CALCULATE INDUCED PRESSURE INCREMENT
 500  SKIN   =8./3.*M*LAMBDA*EL/SURF(I,2)*((1.+SURF(I,5)+SQRT(SURF(I,5)))
     1 /((1.+SQRT(SURF(I,5)))*(1.+TAPER2))-(1.+TAPER1+SQRT(TAPER1))/
     2 ((1.+SQRT(TAPER1))*(1.+TAPER2))*SQRT(ELO/EL))
C
    1 RETURN
C
      END
```

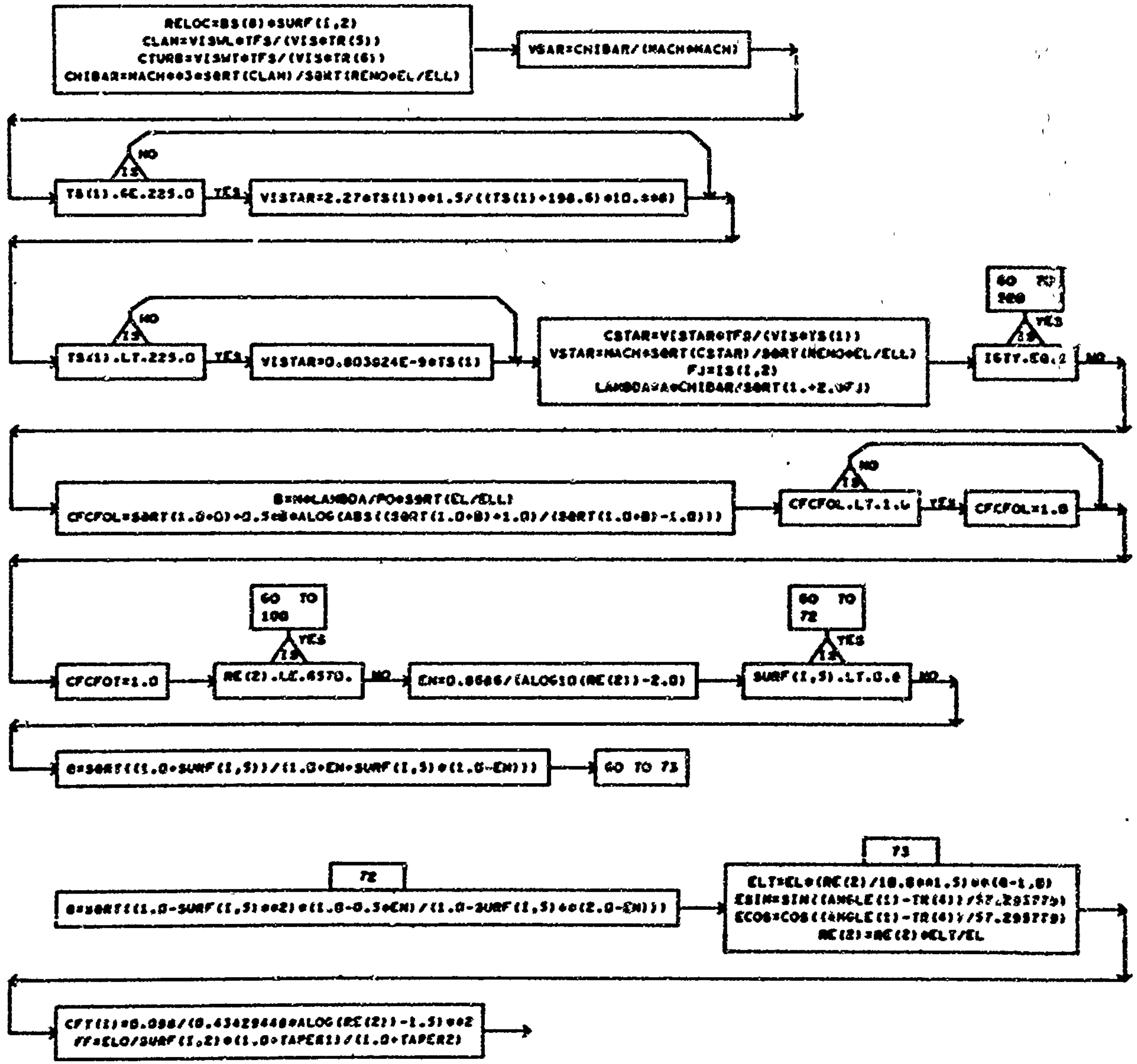



131
SURF(I,5)=EL/EL1
EL=EL1
GO TO 133
132
SURF(I,3)=EL1/EL
133
CONTINUE
TAPER1.EQ.0.0
IS
NO
YES
TAPER1=0.0001
110
CONTINUE
SURF(I,3).EQ.0.0
IS
NO
YES
SURF(I,3)=0.0001
TR(5)=SURF(I,6)
TR(6)=SURF(I,7)
SURF(I,5).LT.0.8
IS
YES
GO TO 77
NO
ELL=EL*4.0*((1.0+SURF(I,5))/(3.0+SURF(I,5)))**2
GO TO 79
77
ELL=EL*(0.75*(1.0-SURF(I,5)**2)/(1.0-SURF(I,5)**1.5))**2
79
RENO=FS(8)*ELL
IS3=IS(I,3)
IS(I,5).NE.1
IS
YES
GO TO 14
NO
ANGLE(1)=SURF(I,4)
GO TO 90
14
COMPUTED GO TO
IF THE VALUE OF IS3 IS
TRANSFER TO STATEMENT
1 21
2 16
3 21
4 16
5 21
6 16
21
ANGLE(1)=SURF(I,3)+TR(4)-SURF(I,4)
IGTYPE.EQ.2
IS
NO
YES
ANGLE(1)=-DELTA
ANGLE(1)
IF
-
GO TO 18
+
GO TO 17
IF
0
GO TO 19
16
ANGLE(1)=SURF(I,3)+TR(4)+SURF(I,4)

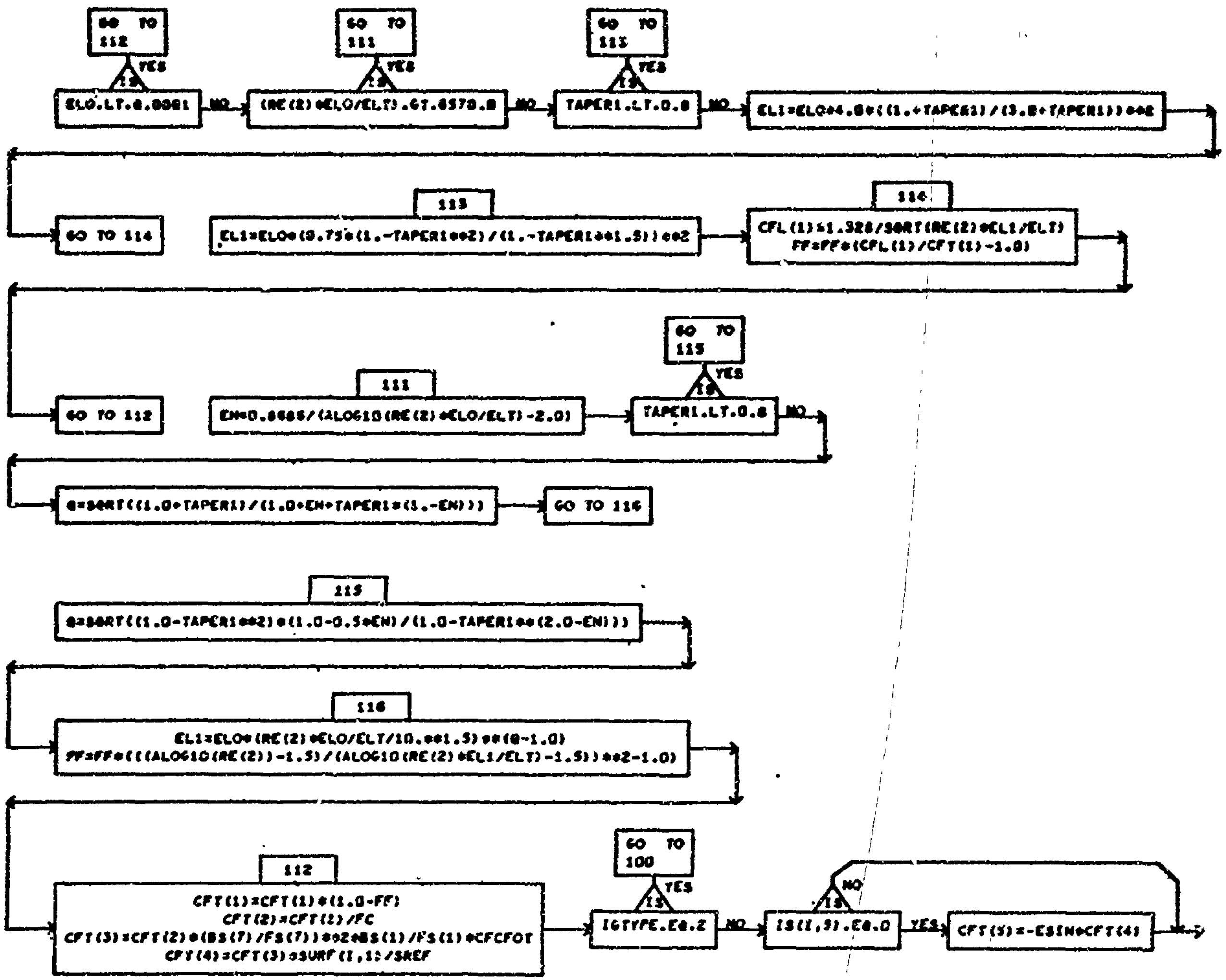

IGTYPE.EQ.2
IS
NO
YES
ANGLE(1)=DELTA
30
CONTINUE
ANGLE(1)
IF
GO TO 17
GO TO 19
10
KF=2
IS(1,2).NE.1
IS
YES
NO
GO TO 86
PERFORM SUBROUTINE
CONE
ANGLE,CP,0
GO TO 24
86
ANGLE(2)=ABS(ANGLE(1))
IS(1,3).LT.5
GO TO 23
KF=5
GO TO 85
23
ISE=0
ISDET=0
PERFORM SUBROUTINE
COMPR
ANGLE,MER,IS(1,9),CPSTAG,TFS,PFS,ISDET,IFIRST,CP
GO TO 87
17
ANGLE(2)=ABS(ANGLE(1))
IS(1,3).LT.3
GO TO 22
KF=4
ANGLE(2)=-ANGLE(2)
GO TO 85
22
KF=1
ISDET=0
PERFORM SUBROUTINE
EXPAND
ANGLE,MER,IS(1,9),ISDET,CP
IGTY.EQ.0.OR.MER.LE.1
GO TO 87
SKIN=0.0
GO TO 1
19
ANGLE(2)=0.0
IS(1,3).LT.5
GO TO 84
KF=6
GO TO 85
84
KF=3
REPEAT TO 20
FOR
K=1,1+1,...,8
20
BS(K)=FS(K)
ONEOM=1.0/BS(6)
ANGLE(3)=ARSIN(ONEOM)*0.572957BE02
GO TO 24
85
ISE=0

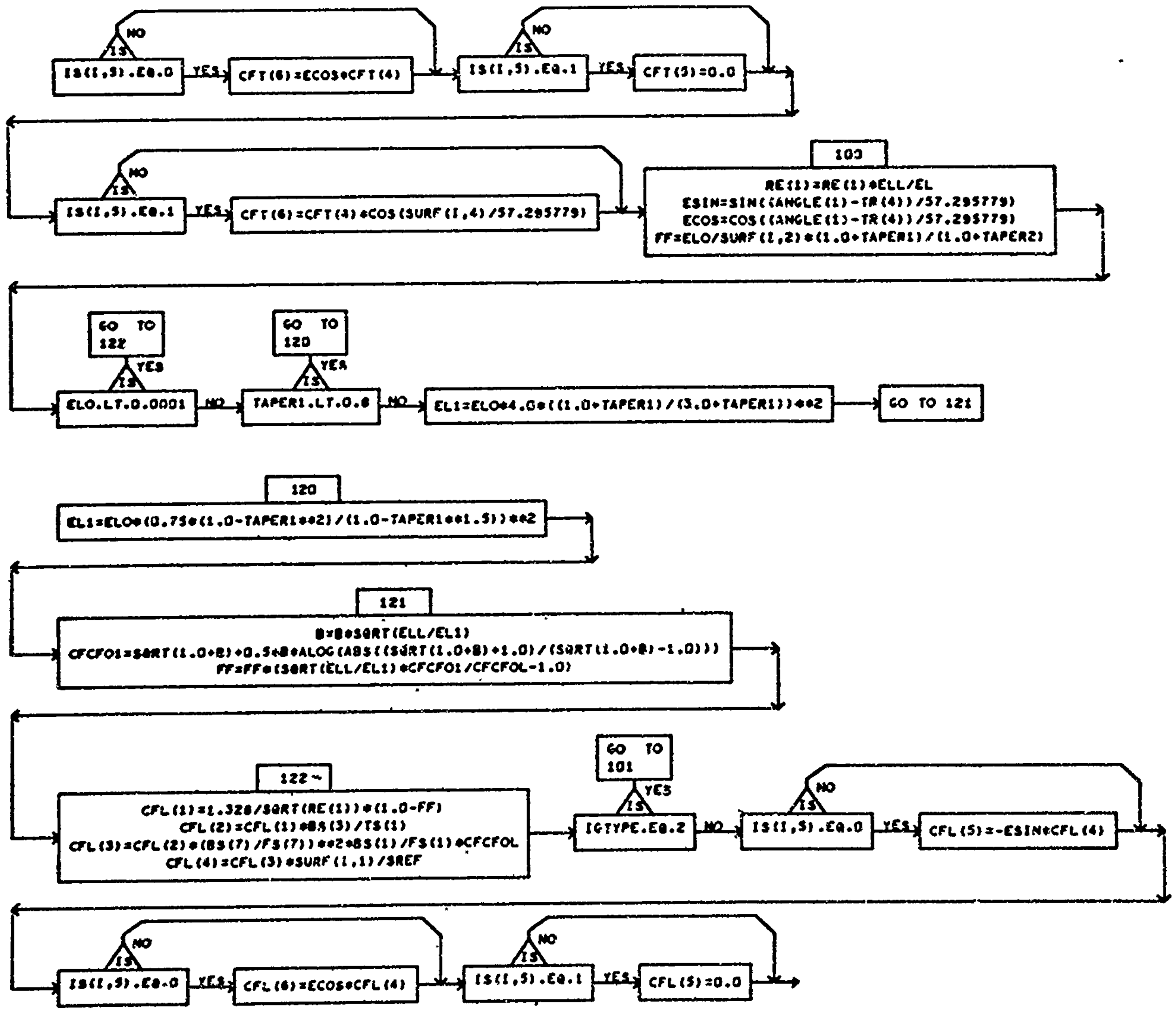

PERFORM SUBROUTINE
NEWTPN
ANGLE,EMN,CP,SURF(I,8),IS(I,9),MER,CPSTAG,TFS,PFS,ISE,IFIRST
GO TO 87
IS
YES
MER.EQ.0
NO
ISDET=1
PERFORM SUBROUTINE
COMPR
ANGLE,MER,IPRINT,CPSTAG,TFS,PFS,ISDET,IFIRST,CP
87
CONTINUE
GO TO 24
IS
YES
MER.LE.1
NO
MEREXP=1
5
MER=1
GO TO 05
24
CONTINUE
GO TO 26
IS
YES
IS(I,9).EQ.0
NO
GO TO 25
IS
YES
IS(I,9).EQ.1
NO
GO TO 26
IS
YES
(NPRT.LT.52).AND.(NS.NE.1)
NO
25
PERFORM SUBROUTINE
HEADER
81
WRITE
WRITE(6,31)ALPHA,MACH,TR(3),ALT,FS(8),SREF
NPRT=8
26
PERFORM SUBROUTINE
TEMP
EL,TR,RE,TS,IS(I,8),MER,IS(I,9),RT
GO TO 83
IF
0
MER-1
-
IF
+
GO TO 5
83
KO=MACH*SIN(ANGLE(1)/57.295779)
IS
NO
IS3.EQ.1.OR.IS3.EQ.3.OR.IS3.EQ.5
YES
KO=-KO

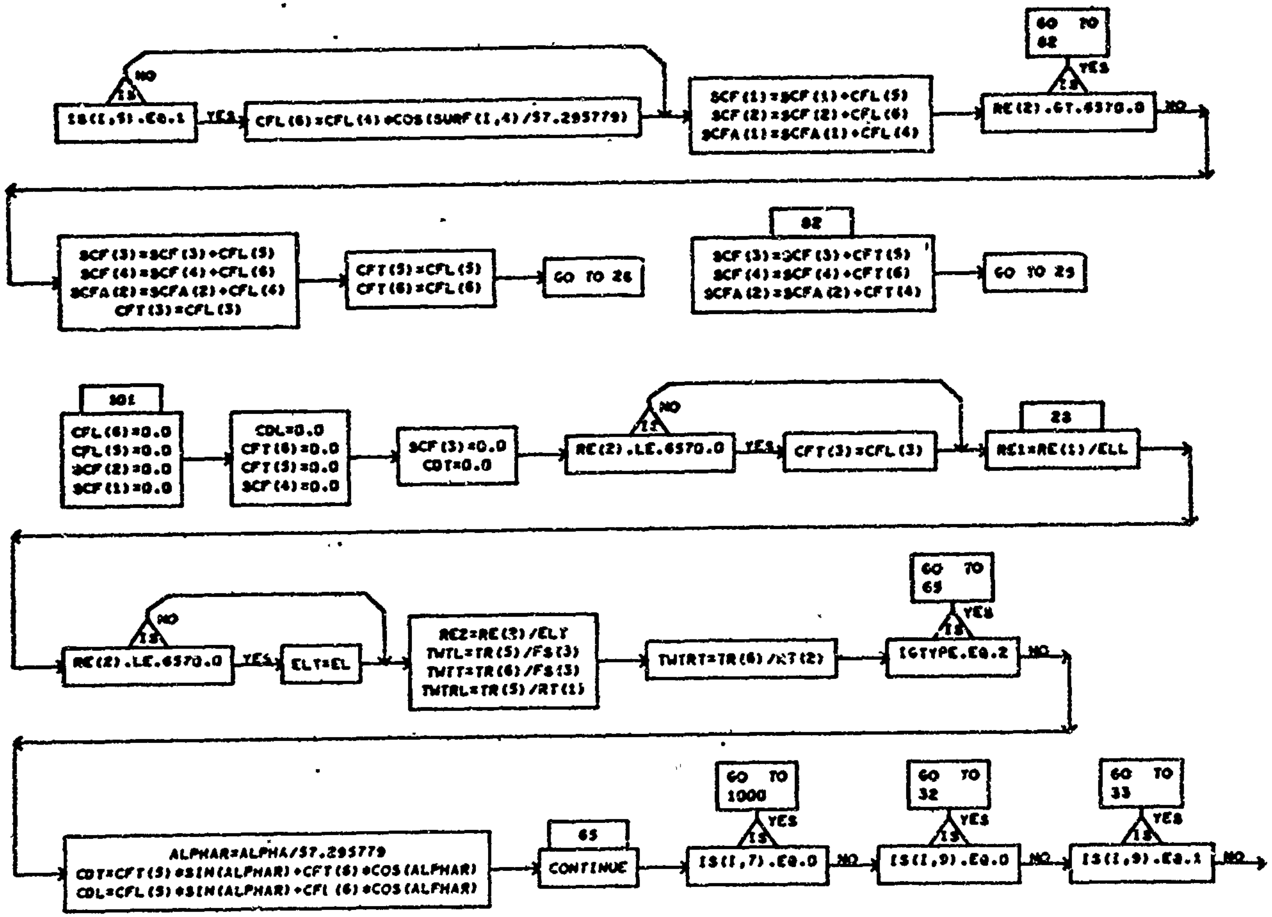

GO TO 91
IS
YES
KO.LT.-0.0001.OR.KO.GT.0.0001
NO
PO=1.0
M=1.9156
GO TO 92
91
GA=1.4
GO TO 93
IS
YES
KO.GE.0.0
NO
PO=0.00001
M=0.00021
GO TO 92
IS
YES
KO.LE.-2./(GA-1.)
NO
PO=(1.0+0.5*(GA-1.)*KO)**(2.*GA/(GA-1.))
GO TO 94
IS
YES
KO.GE.-3.0
NO
M=1.426+0.219*KO
GO TO 92
93
PO=1.0+0.25*GA*(GA+1.0)*KO*KO+GA*KO*SQRT(1.0+(0.25*(GA+1.0)*KO)**2)
94
M=1.9156+KO*(0.41727-KO*(0.0419101+KO*(0.010427-KO*(0.00214381-KO*1.03217E-4))
))
92
TWTRL=TR(5)/RT(1)
G=0.34456*(TW.RL+0.3838)
A=G/2.0
IS
NO
TR(5).GE.225.0
YES
VISWL=2.27*TR(5)**1.5/((TR(5)+198.6)*10.**8)
IS
NO
TR(6).GE.225.0
YES
VISWT=2.27*TR(6)**1.5/((TR(6)+198.6)*10.**8)
IS
NO
TR(5).LT.225.0
YES
VISWL=0.80382436E-9*TR(5)
IS
NO
TR(6).LT.225.0
YES
VISWT=0.80382436E-9*TR(6)

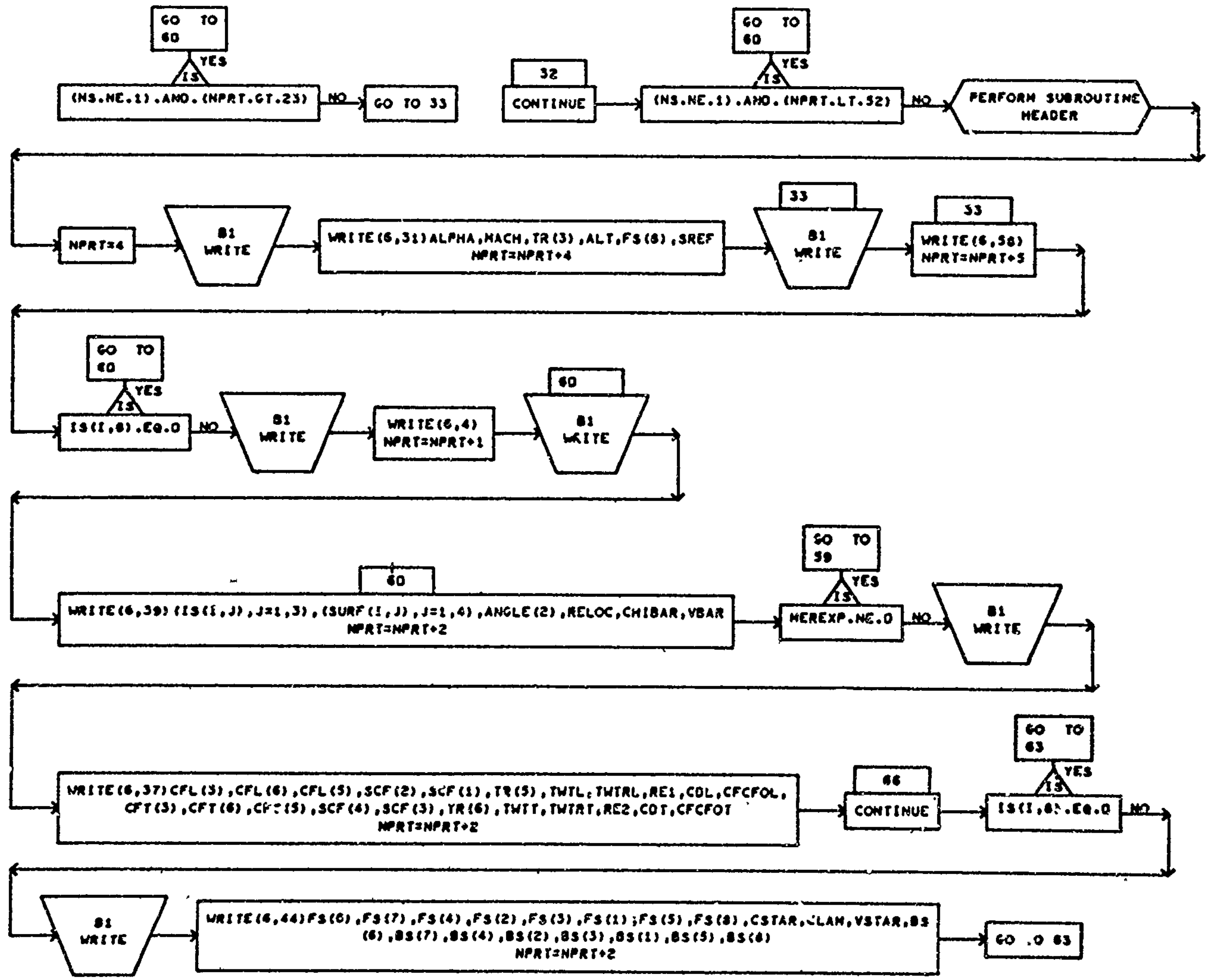


RELOC=BS(8)*SURF(I,2)
CLAM=VISML*TFS/(VIS*TR(5))
CTURB=VISMT*TFS/(VIS*TR(6))
CHIBAR=MACH**3*SQRT(CLAM)/SQRT(RENO*EL/ELL)
VSAR=CHIBAR/(MACH*MACH)
TS(1).GE.225.0
IS
NO
YES
VISTAR=2.27*TS(1)**1.5/((TS(1)+198.6)*10.**8)
TS(1).LT.225.0
IS
NO
YES
VISTAR=0.803024E-9*TS(1)
CSTAR=VISTAR*TFS/(VIS*TS(1))
VSTAR=MACH*SQRT(CSTAR)/SQRT(RENO*EL/ELL)
FJ=IS(I,2)
LAMBDA=A*CHIBAR/SQRT(1.+2.*FJ)
GO TO 200
YES
IS
ISTY.EQ.2
NO
B=N*LAMBDA/PO*SQRT(EL/ELL)
CFCFOL=SQRT(1.0+O)+0.5*B*ALOG(ABS((SQRT(1.0+B)+1.0)/(SQRT(1.0+B)-1.0)))
CFCFOL.LT.1.0
IS
NO
YES
CFCFOL=1.0
CFCFOT=1.0
GO TO 100
YES
IS
RE(2).LE.6570.
NO
EN=0.8686/(ALOG10(RE(2))-2.0)
GO TO 72
YES
IS
SURF(I,5).LT.0.0
NO
G=SQRT((1.0+SURF(I,5))/(1.0+EN*SURF(I,5)*(1.0-EN)))
GO TO 73
72
G=SQRT((1.0-SURF(I,5)**2)*(1.0-0.5*EN)/(1.0-SURF(I,5)**(2.0-EN)))
73
ELT=EL*(RE(2)/10.0**1.5)**(G-1.0)
ESIN=SIN((ANGLE(I)-TR(4))/57.295779)
ECOS=COS((ANGLE(I)-TR(4))/57.295779)
RE(2)=RE(2)*ELT/EL
CFT(I)=0.088/(0.43429448*ALOG(RE(2))-1.5)**2
FF=ELO/SURF(I,2)*(1.0+TAPER1)/(1.0+TAPER2)


GO TO 112
IS
YES
ELO.LT.0.0001
NO
GO TO 111
(RE(2)*ELO/ELT).GT.6570.0
GO TO 113
TAPER1.LT.0.8
ELI=ELO*4.0*((1.+TAPER1)/(3.0+TAPER1))**2
GO TO 114
113
ELI=ELO*(0.75*(1.-TAPER1**2)/(1.-TAPER1**1.5))**2
114
CFL(1)=1.328/SQRT(RE(2)*ELI/ELT)
FF=FF*(CFL(1)/CFT(1)-1.0)
GO TO 112
111
EN=0.8685/(ALOG10(RE(2)*ELO/ELT)-2.0)
GO TO 115
TAPER1.LT.0.8
Q=SQRT((1.0+TAPER1)/(1.0+EN+TAPER1*(1.-EN)))
GO TO 116
115
Q=SQRT((1.0-TAPER1**2)*(1.0-0.5*EN)/(1.0-TAPER1**(2.0-EN)))
116
ELI=ELO*(RE(2)*ELO/ELT/10.**1.5)**(Q-1.0)
FF=FF*(((ALOG10(RE(2))-1.5)/(ALOG10(RE(2)*ELI/ELT)-1.5))**2-1.0)
112
CFT(1)=CFT(1)*(1.0-FF)
CFT(2)=CFT(1)/FC
CFT(3)=CFT(2)*(BS(7)/FS(7))**2*BS(1)/FS(1)*CFCFOT
CFT(4)=CFT(3)*SURF(I,1)/SREF
GO TO 100
IGTYPE.EQ.2
IS(I,5).EQ.0
CFT(5)=-ESIN*CFT(4)

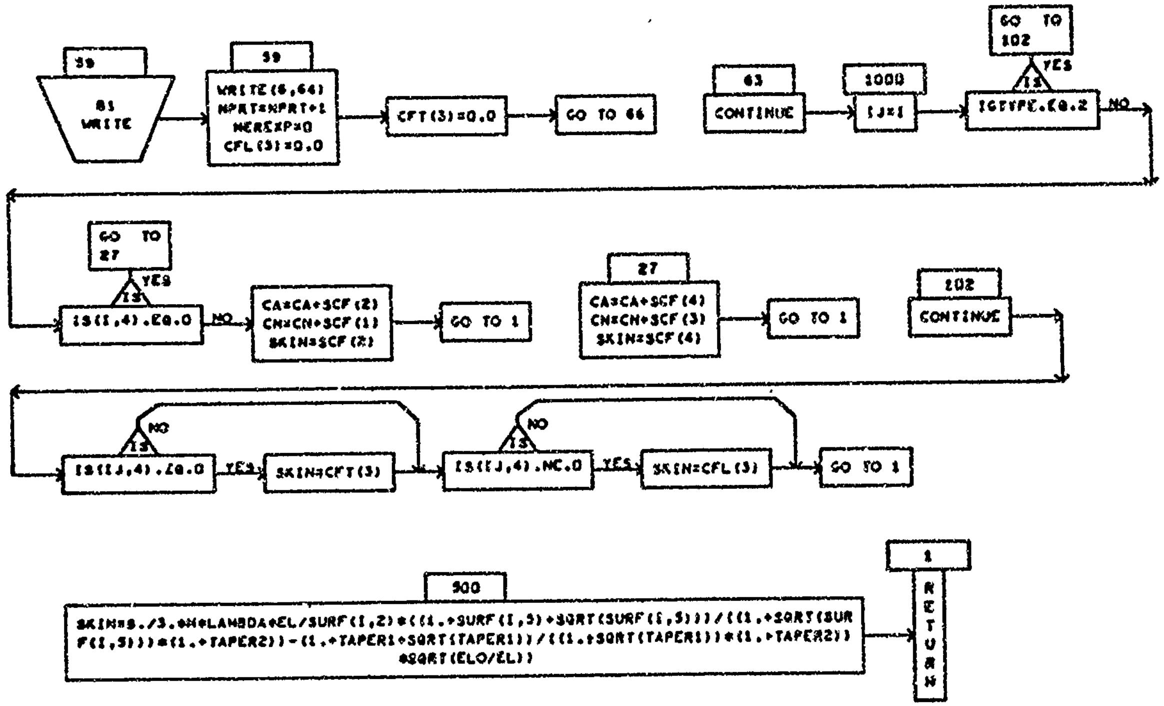

IS(I,5).EQ.0
YES
NO
CFT(6)=ECOS*CFT(4)
IS(I,5).EQ.1
CFT(5)=0.0
IS(I,5).EQ.1
CFT(6)=CFT(4)*COS(SURF(I,4)/57.295779)
103
RE(1)=RE(1)*ELL/EL
ESIN=SIN((ANGLE(1)-TR(4))/57.295779)
ECOS=COS((ANGLE(1)-TR(4))/57.295779)
FF=ELO/SURF(I,2)*(1.0+TAPER1)/(1.0+TAPER2)
GO TO 122
ELO.LT.0.0001
GO TO 120
TAPER1.LT.0.8
EL1=ELO*4.0*((1.0+TAPER1)/(3.0+TAPER1))**2
GO TO 121
120
EL1=ELO*(0.75*(1.0-TAPER1**2)/(1.0-TAPER1**1.5))**2
121
B=B*SQRT(ELL/EL1)
CFCFO1=SQRT(1.0+B)+0.5*B*ALOG(ABS((SQRT(1.0+B)+1.0)/(SQRT(1.0+B)-1.0)))
FF=FF*(SQRT(ELL/EL1)*CFCFO1/CFCFOL-1.0)
122
CFL(1)=1.328/SQRT(RE(1))*(1.0-FF)
CFL(2)=CFL(1)*BS(3)/TS(1)
CFL(3)=CFL(2)*(BS(7)/FS(7))**2*BS(1)/FS(1)*CFCFOL
CFL(4)=CFL(3)*SURF(1,1)/SREF
GO TO 101
IGTYPE.EQ.2
IS(I,5).EQ.0
CFL(5)=-ESIN*CFL(4)
IS(I,5).EQ.0
CFL(6)=ECOS*CFL(4)
IS(I,5).EQ.1
CFL(5)=0.0


IS(I,5).EQ.1
IS
NO
YES
CFL(6)=CFL(4)*COS(SURF(I,4)/57.295779)
SCF(1)=SCF(1)+CFL(5)
SCF(2)=SCF(2)+CFL(6)
SCFA(1)=SCFA(1)+CFL(4)
RE(2).GT.6570.0
IS
YES
GO TO 82
NO
SCF(3)=SCF(3)+CFL(5)
SCF(4)=SCF(4)+CFL(6)
SCFA(2)=SCFA(2)+CFL(4)
CFT(3)=CFL(3)
CFT(5)=CFL(5)
CFT(6)=CFL(6)
GO TO 26
82
SCF(3)=SCF(3)+CFT(5)
SCF(4)=SCF(4)+CFT(6)
SCFA(2)=SCFA(2)+CFT(4)
GO TO 25
101
CFL(6)=0.0
CFL(5)=0.0
SCF(2)=0.0
SCF(1)=0.0
CDL=0.0
CFT(6)=0.0
CFT(5)=0.0
SCF(4)=0.0
SCF(3)=0.0
CDT=0.0
RE(2).LE.6570.0
IS
NO
YES
CFT(3)=CFL(3)
23
RE1=RE(1)/ELL
RE(2).LE.6570.0
IS
NO
YES
ELT=EL
RE2=RE(3)/ELT
TWTL=TR(5)/FS(3)
TWTT=TR(6)/FS(3)
TWTRL=TR(5)/RT(1)
TWTRT=TR(6)/RT(2)
IGTYPE.EQ.2
IS
YES
GO TO 65
NO
ALPHAR=ALPHA/57.295779
CDT=CFT(5)*SIN(ALPHAR)+CFT(6)*COS(ALPHAR)
CDL=CFL(5)*SIN(ALPHAR)+CFL(6)*COS(ALPHAR)
65
CONTINUE
IS(I,7).EQ.0
IS
YES
GO TO 1000
NO
IS(I,9).EQ.0
IS
YES
GO TO 32
NO
IS(I,9).EQ.1
IS
YES
GO TO 33
NO



GO TO 60
YES
IS
(NS.NE.1).AND.(NPRT.GT.23)
NO
GO TO 33
32
CONTINUE
GO TO 60
YES
IS
(NS.NE.1).AND.(NPRT.LT.52)
NO
PERFORM SUBROUTINE HEADER
NPRT=4
B1 WRITE
WRITE(6,31)ALPHA,MACH,TR(3),ALT,FS(8),SREF
NPRT=NPRT+4
33
B1 WRITE
33
WRITE(6,58)
NPRT=NPRT+5
GO TO 60
YES
IS
IS(I,8).EQ.0
NO
B1 WRITE
WRITE(6,4)
NPRT=NPRT+1
60
B1 WRITE
60
WRITE(6,39)(IS(I,J),J=1,3),(SURF(I,J),J=1,4),ANGLE(2),RELOC,CHIBAR,VBAR
NPRT=NPRT+2
GO TO 59
YES
IS
NEREXP.NE.0
NO
B1 WRITE
WRITE(6,37)CFL(3),CFL(6),CFL(5),SCF(2),SCF(1),TR(5),TWTL,TWTRL,RE1,CDL,CFCFOL,
CFT(3),CFT(6),CF:(5),SCF(4),SCF(3),TR(6),TWTT,TWTRT,RE2,CDT,CFCFOT
NPRT=NPRT+2
66
CONTINUE
GO TO 63
YES
IS
NO
B1 WRITE
WRITE(6,44)FS(6),FS(7),FS(4),FS(2),FS(3),FS(1);FS(5),FS(8),CSTAR,CLAM,VSTAR,BS
(6),BS(7),BS(4),BS(2),BS(3),BS(1),BS(5),BS(8)
NPRT=NPRT+2
GO .O 63

SKINFR



59
81 WRITE
59
WRITE (6,64)
NPRT=NPRT+1
NERSXP=0
CFL(3)=0.0
CFT(3)=0.0
GO TO 66
63
CONTINUE
1000
IJ=1
GO TO 102
YES
IS
IGTYPE.EQ.2
NO
GO TO 27
YES
IS
IS(I,4).EQ.0
NO
CA=CA+SCF(2)
CN=CN+SCF(1)
SKIN=SCF(2)
GO TO 1
27
CA=CA+SCF(4)
CN=CN+SCF(3)
SKIN=SCF(4)
GO TO 1
202
CONTINUE
NO
IS
IS(IJ,4).EQ.0
YES
SKIN=CFT(3)
NO
IS
IS(IJ,4).NE.0
YES
SKIN=CFL(3)
GO TO 1
300
SKIN=8./3.*H*LAMBDA*EL/SURF(I,2)*((1.+SURF(I,3)+SQRT(SURF(I,3)))/((1.+SQRT(SURF(I,3)))*(1.+TAPER2))-(1.+TAPER1+SQRT(TAPER1))/((1.+SQRT(TAPER1))*(1.+TAPER2))
*SQRT(ELO/EL))
1
RETURN

SYMBOLS USED IN SUBROUTINE SKINFR

A	R	U	SPEED OF SOUND	SKINFR
AFS	R	A	FREE-STREAM SPEED OF SOUND	SKINFR
ALP	R	C	ANGLE OF ATTACK ARRAY	SKINFR
ALPHA	R	A	ANGLE OF ATTACK, DEGREES	SKINFR
ALT	R	A	ALTITUDE, FEET	SKINFR
ANGLE	R	D	FLOW ANGLE ARRAY	SKINFR
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	SKINFR
B	R	U	LAMINAR VISCOUS INTERACTION PARAMETER	SKINFR
BET	R	C	YAW ANGLE ARRAY	SKINFR
BS	R	C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	SKINFR
CA	R	A	AXIAL FORCE COEFFICIENT	SKINFR
CASE	I	C	CASE NUMBER	SKINFR
CCA	R	C	AXIAL FORCE COEFFICIENT ARRAY	SKINFR
CCD	R	C	DRAG COEFFICIENT ARRAY	SKINFR
CCL	R	C	LIFT COEFFICIENT ARRAY	SKINFR
CCLL	R	C	ROLLING MOMENT COEFFICIENT ARRAY	SKINFR
CCLM	R	C	PITCHING MOMENT COEFFICIENT ARRAY	SKINFR
CCLN	R	C	YAWING MOMENT COEFFICIENT ARRAY	SKINFR
CCN	R	C	NORMAL FORCE COEFFICIENT ARRAY	SKINFR
CCY	R	C	SIDE FORCE COEFFICIENT ARRAY	SKINFR
CDL	R	U	LAMINAR FLOW DRAG COEFFICIENT	SKINFR
CDT	R	U	TURBULENT FLOW DRAG COEFFICIENT	SKINFR
CF	R	C	SKIN FRICTION CONTRIBUTION TO AXIAL FORCE ARRAY	SKINFR
CFCFOL	R	U	LAMINAR RATIO OF SKIN FRACTION, INTERACTION/NO INTERACTION	SKINFR
CFCFOT	R	U	TURBULENT RATIO OF SKIN FRICTION = 1.0	SKINFR
CFCFOI	R	U	INITIAL SURFACE SKIN FRICTION RATIO	SKINFR
CFL	R	D	LAMINAR SKIN FRICTION COEFFICIENTS	SKINFR
CFT	R	D	TURBULENT SKIN FRICTION COEFFICIENTS	SKINFR
CHIBAR	R	U	HYPERSONIC INTERACTION PARAMETER	SKINFR
CKU	R	C	LAMINAR FLOW FLIGHT CONDITION CONSTANT	SKINFR
CLAM	R	U	CHAPMAN-RUBESIN VISCOSITY COEFFICIENT, LAMINAR	SKINFR
CLOD	R	C	LIFT TO DRAG RATIO ARRAY	SKINFR
CN	R	A	NORMAL FORCE COEFFICIENT	SKINFR
CP	R	U	PRESSURE COEFFICIENT	SKINFR
CPS	R	C	ARRAY FOR NEWTONIAN CORRELATION FACTOR, K	SKINFR
CPSTAG	R	A	MODIFIED NEWTONIAN CORRELATION FACTOR, K	SKINFR

SYMBOLS USED IN SUBROUTINE SKINFR

CSTAR	R	U	LINEAR VISCOSITY COEFFICIENT AT REFERENCE CONDITION	SKINFR
CTURB	R	U	CHAPMAN-RUBESIN VISCOSITY COEFFICIENT, TURBULENT	SKINFR
DELTA	R	A	IMPACT ANGLE, DEGREES	SKINFR
ECOS	R	U	COSINE OF FLOW TURNING ANGLE	SKINFR
EL	R	U	SURFACE REFERENCE LENTH, INPUT	SKINFR
ELL	R	U	EFFECTIVE SURFACE LENGTH, LAMINAR	SKINFR
ELLOC	R	C	REFERENCE LENGTH (=EL)	SKINFR
ELO	R	U	LENGTH OF INITIAL SURFACE	SKINFR
ELT	R	U	EFFECTIVE SURFACE LENGTH, TURBULENT	SKINFR
EL1	R	U	EFFECTIVE LENGTH OF INITIAL SURFACE	SKINFR
EMN	R	U	MACH NUMBER TIMES SHOCK ANGLE SQUARED	SKINFR
EN	R	U	PARAMETER IN CHARACTERISTIC LENGTH EQUATION	SKINFR
ERROR	I	C	ERROR FLAG	SKINFR
ESIN	R	U	SINE OF FLOW TURNING ANGLE	SKINFR
ETACS	R	C	SAVED VALUES OF PRANDTL-MEYER CORRECTION FACTOR	SKINFR
FC	R	C	TURBULENT FLOW, SKIN FRICTION COMPRESSIBILITY FACTOR	SKINFR
FF	R	U	FRICTION FACTOR	SKINFR
FJ	R	U	MANGLER TRANSFORMATION PARAMETER	SKINFR
FRX	R	C	TURBULENT FLOW, REYNOLDS NUMBER COMPRESSIBILITY FACTOR	SKINFR
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	SKINFR
G	R	U	RATIO OF SPECIFIC HEATS	SKINFR
GCP	R	C	GAS SPECIFIC HEAT AT CONSTANT PRESSURE	SKINFR
HAW	R	C	ADIABATIC-WALL ENTHALPY	SKINFR
HW	R	C	WALL ENTHALPY	SKINFR
H1	R	C	FREE-STREAM ENTHALPY	SKINFR
H2	R	C	LOCAL ENTHALPY	SKINFR
I	I	U	DO-LOOP INDEX (SKIN FRICTION SURFACE NUMBER)	SKINFR
IFIRST	I	A	INITIAL POINT FLAG FOR NEWTPM	SKINFR
IGTY	I	U	INDUCED PRESSURE FLAG	SKINFR
IGTYPE	I	A	GEOMETRY TYPE	SKINFR
II	I	U	DO-LOOP INDEX (SKIN FRICTION SURFACE NUMBER)	SKINFR
IM	I	C	ELEMENT ROW NUMBER ARRAY	SKINFR
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	SKINFR
IPRINT	I	U	PRINT FLAG	SKINFR
IS	I	C	SKIN FRICTION FLAG DATA ARRAY	SKINFR
ISDET	I	U	DATA GENERATION CONTROL FLAG	SKINFR

SYMBOLS USED IN SUBROUTINE SKINFR

ISE	I	U	DATA GENERATION CONTROL FLAG	SKINFR
ISIZ	I	A	NUMBER OF ELEMENTS STORED IN CORE	SKINFR
IS3	I	U	PRESSURE CALCULATION METHOD FLAG	SKINFR
KF	I	U	METHOD-SURFACE TYPE FLAG	SKINFR
KO	R	U	SIMILARITY PARAMETER	SKINFR
L	I	C	NUMBER OF ELEMENTS	SKINFR
LAMBDA	R	U	MODIFIED HYPERSONIC INTERACTION PARAMETER	SKINFR
LS	I	C	NUMBER OF ELEMENTS	SKINFR
M	R	U	SLOPE OF PRESSURE RATIO VERSUS LAMBDA	SKINFR
MACH	R	A	MACH NUMBER	SKINFR
MER	I	U	ERROR FLAG	SKINFR
MEREXP	I	U	VACUUM EXPANSION FLAG	SKINFR
NC	I	U	NOT USED	SKINFR
NPRT	I	C	PRINT COUNTER	SKINFR
NS	I	A	NUMBER OF SKIN FRICTION SURFACES	SKINFR
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	SKINFR
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	SKINFR
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	SKINFR
ONEOM	R	U	1.0 DIVIDED BY MACH NUMBER	SKINFR
PAGE	I	C	PAGE NUMBER	SKINFR
PFS	R	A	FREE-STREAM PRESSURE-LBS/SQUARE FOOT	SKINFR
PO	R	U	PRESSURE RATIO AT LAMBDA = 0.0	SKINFR
Q	R	U	TAPER RATIO CORRECTION EQUATION EXPONENT	SKINFR
RE	R	D	REFERENCE REYNOLDS NUMBER	SKINFR
RELOC	R	U	LOCAL REYNOLDS NUMBER	SKINFR
RENO	R	U	FREE-STREAM REYNOLDS NUMBER	SKINFR
RET	R	C	TURBULENT FLOW REYNOLDS NUMBER AT REFERENCE CONDITION	SKINFR
RHOFS	R	A	FREE STREAM DENSITY	SKINFR
ROMURA	R	C	SQUARE-ROOT OF REFERENCE DENSITY-VISCOSITY RATIO	SKINFR
RT	R	D	RECOVERY TEMPERATURE	SKINFR
SCF	R	D	TOTAL SKIN FRICTION	SKINFR
SCFA	R	D	TOTAL SKIN FRICTION	SKINFR
SKIN	R	A	TOTAL AXIAL SKIN FRICTION CONTRIBUTION	SKINFR
SREF	R	A	VEHICLE REFERENCE AREA (WING AREA)	SKINFR
SURF	R	C	SKIN FRICTION DATA ARRAY	SKINFR
TAPER1	R	U	TAPER RATIO OF INITIAL SURFACE	SKINFR

SKINFR

SYMBOLS USED IN SUBROUTINE SKINFR

TAPER2	R	U	TAPER RATIO OF SURFACE	SKINFR
TFS	R	A	FREE-STREAM TEMPERATURE	SKINFR
TITLE	R	C	TITLE	SKINFR
TR	R	D	FLIGHT CONDITION AND SKIN FRICTION DATA ARRAY	SKINFR
TS	R	D	REFERENCE TEMPERATURE (T STAR)	SKINFR
TST1	R	C	REFERENCE TO FREE-STREAM TEMPERATURE (OR ENTHALPY) RATIO	SKINFR
TWTL	R	U	WALL TO FREE-STREAM TEMPERATURE RATIO, LAMINAR	SKINFR
TWTRL	R	U	WALL TO RECOVERY TEMPERATURE RATIO, LAMINAR	SKINFR
VBAR	R	U	HYPERSONIC VISCOUS PARAMETER	SKINFR
VIS	R	A	FREE STREAM VISCOSITY	SKINFR
VISTAR	R	U	VICOSITY AT REFERENCE CONDITION	SKINFR
VISWL	R	U	VISCOSITY AT WALL TEMPERATURE, LAMINAR	SKINFR
VISWT	R	U	VISCOSITY AT WALL TEMPERATURE, TURBULENT	SKINFR
VSTAR	R	U	VICOUS-INTERACTION PARAMETER	SKINFR
XCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-X	SKINFR
YCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Y	SKINFR
ZCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Z	SKINFR

14. SUBROUTINE COMPR (DECK AROM)

This routine calculates the local flow properties using conventional oblique-shock relationships (NACA TR 1135).

a. Algorithm

First the constants in the oblique equation are calculated. A check is made for shock detachment, and if the shock is not detached the three real roots for the cubic are found. If the shock has detached, the conditions will be calculated by the NEWTPM routine. The proper root is selected and local flow conditions are calculated.

b. Input/Output

None

c. Error

An error condition will occur if a negative value is found for sine theta squared. Set it to 0 and the program will continue.

d. Subroutines Required

NEWTPM

e. Argument List

(ANGLE, MER, IPRINT, CPSTAG, TFS, PFS, ISDET, IFIRST, CP)

f. Length

2794 bytes

DECK AROM

```
      SUBROUTINE COMPR (ANGLE,MER,IPRINT,CPSTAG,TFS,PFS,ISDET,IFIRST,CP)
C
C  USING THE FREE STREAM MACH NUMBER AND THE EQUIVALENT WEDGE ANGLE,
C  THIS ROUTINE COMPUTES THE CONDITIONS BEHIND THE SHOCK
C
      DIMENSION  FS(8),ANGLE(3),BS(8),R(3)
      DIMENSION  TITLE(15)
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),
     1  YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)
      COMMON  CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1  AREA2,IN,IM,K,LS,FS,BS
C
      INTEGER  CASE,PAGE,ERROR
      REAL  NX2, NY2, NZ2
      REAL        MACHSQ, MACHO, MACH1
      MER = 0
      IF (ISDET .EQ. 1) GO TO 24
      IF (ABS(ANGLE(2)) .LT. 0.00001) GO TO 35
      IF (ANGLE(2) .GT. 55.0) GO TO 1
      IF (ABS(ANGLE(2)) .LE. 2.0) GO TO 37
C
C  SET UP CUBIC TO BE SOLVED FOR SIN**2 THETA (SHOCK ANGLE) - CONSTANTS
C  DEFINED IN EQUATION 150B OF TR 1135
      B = -(FS(6)**2 + 2.0)/FS(6)**2 - 1.4*SIN(ANGLE(2)/57.295779)**2
      C =  (2.0*FS(6)**2 + 1.0)/FS(6)**4 + (1.44 + 0.4/FS(6)**2)*
     1     SIN(ANGLE(2)/57.295779)**2
      D = -COS(ANGLE(2)/57.295779)**2/FS(6)**4

C     CHECK FOR SHOCK DETACHMENT
      IF (((-B**2/9. + C/3.)**3 + ((B/3.)**3 - (B*C - 3.*D)/6.)**2)
     1       .GE. 0.0) GO TO 1

C     SHOCK NOT DETACHED, COMPUTE THREE REAL ROOTS

      Y      = B**2 - 3.*C
```

DECK AROM

```
      Z       = (9.*B*C - 2.*B**3 - 27.*D)/(2.*Y**1.5)
      W       = ARCOS(Z)
      Z       = W/3.
      Y       = 2.*SQRT(Y)
      R(1)    = (Y*COS(Z) - B)/3.
      R(2)    = -(Y*COS(Z + 60./57.295779) + B)/3.
      R(3)    = -(Y*COS(Z - 60./57.295779) + B)/3.
      GO TO 3

C  CUBIC SOLUTION WAS NOT FOUND BECAUSE THE SHOCK HAS DETACHED. FLOW
C  PROPERTIES WILL BE CALCULATED BY THE METHOD OF KAUFMAN
    1 ETAC    = 1.0
      ISE = 0
      CALL NEWTPM (ANGLE,EMN,CP,ETAC,IPRINT,MER,CPSTAG,
     1  TFS,PFS,ISE,IFIRST)
      IF (MER - 1) 17,24,24
C
C  A SOLUTION TO THE CUBIC WAS FOUND. CHECK FOR DESIRED SOLUTION.
C  SMALLEST ROOT REQUIRES A DECREASE IN ENTROPY WHICH IS NOT ALLOWED.
C  LARGEST ROOT IS NOT ATTAINED IN PRACTICAL CASES. THEREFORE PICK
C  MIDDLE ANSWER.
    3 IF (R(1) - R(2)) 4,13,5
    4 K = 1
      GO TO 6
    5 K = 2
    6 IF (R(2) - R(3)) 7,14,8
    7 L = 1
      GO TO 9
    8 L = 2
    9 IF (K .EQ. L) GO TO 14
      IF (R(1) - R(3)) 11,15,12
   11 GO TO (15,13), K
   12 GO TO (13,15), K
   13 ANGLE(3) = R(1)
       GO TO 16
   14 ANGLE(3) = R(2)
```

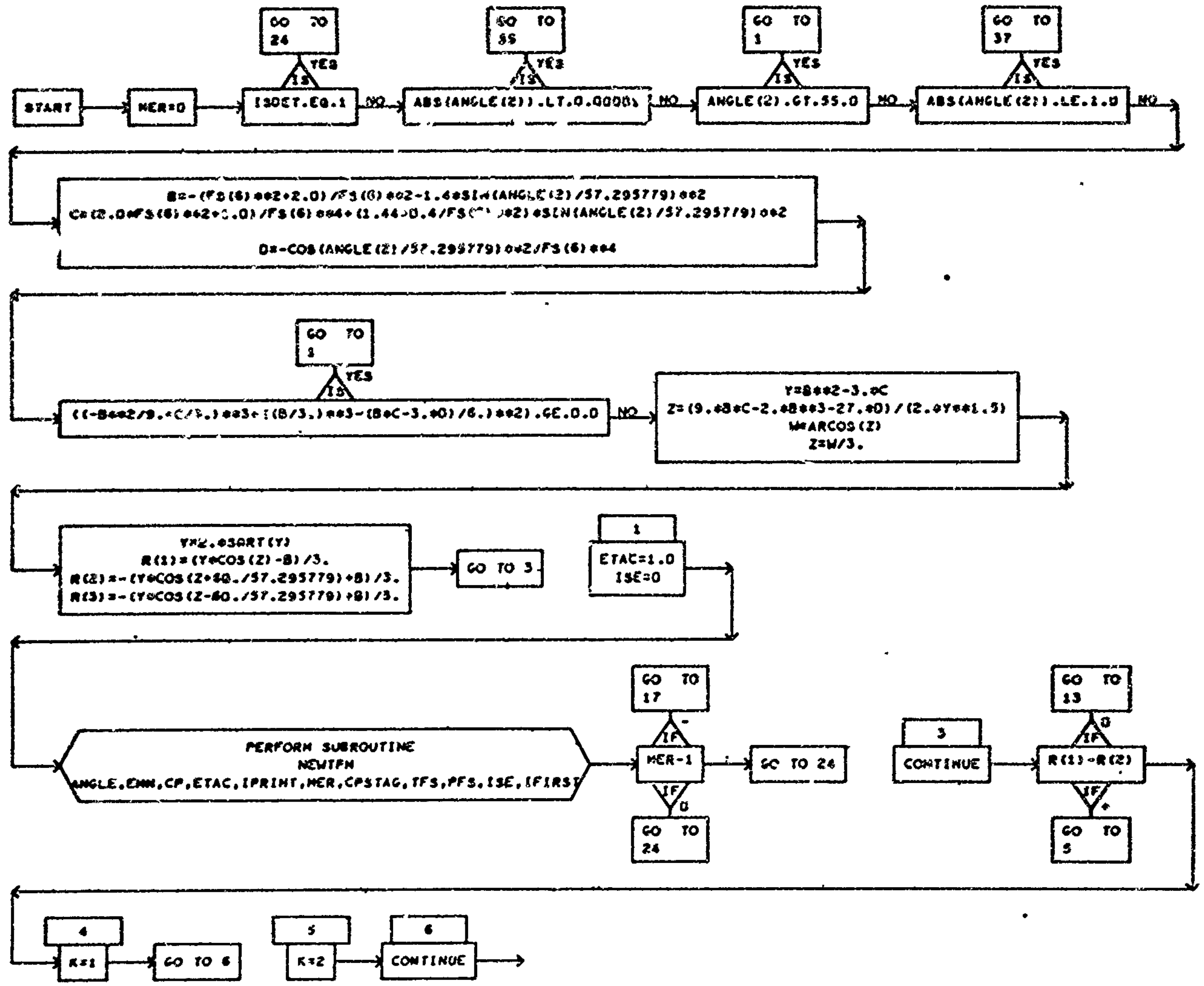
DECK AROM

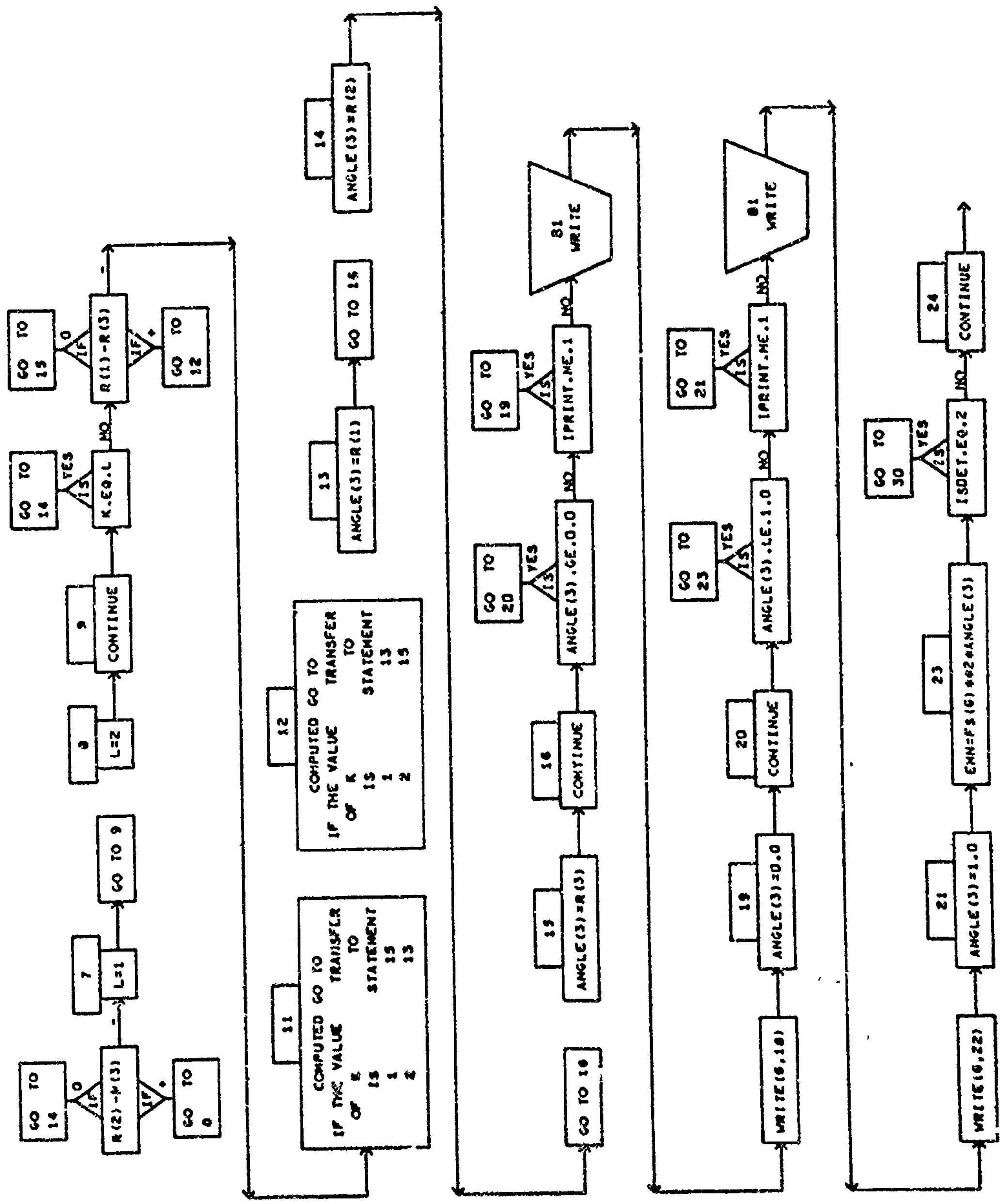
```
      GO TO 16
   15 ANGLE(3) = R(3)
C
C  CHECK IF ANGLE IS NEGATIVE AND PRINT ERROR NOTE IF REQUIRED
   16 IF (ANGLE(3) .GE. 0.0) GO TO 20
      IF (IPRINT .NE. 1) GO TO 19
      WRITE (6,18)
   18 FORMAT (1H0,39H NEGATIVE VALUE FOUND FOR SIN**2 THETA
     1  41HIN CUBIC. TO CONTINUE, IT IS SET TO ZERO.  )
   19 ANGLE(3) = 0.0
   20 IF (ANGLE(3) .LE. 1.0) GO TO 23
      IF (IPRINT .NE. 1) GO TO 21
      WRITE (6,22)
   22 FORMAT (1H0,41H IN CUBIC, SIN**2 THETA GREATER THAN ONE.
     1 31H TO CONTINUE, IT IS SET TO ONE.  )
   21 ANGLE(3) = 1.0
C
C  CALCULATE CONDITIONS BEHIND THE SHOCK USING THE SELECTED SIN**2 THETA
   23 EMN    = FS(6)**2 * ANGLE(3)
      IF (ISDET .EQ. 2) GO TO 30
C  DENSITY         EQ. 129 OF TR 1135
   24 IF (EMN .LT. 1.01) EMN = 1.01
      BS(1) = FS(1)*6.0 * EMN / (EMN + 5.0)
C  PRESSURE        EQ. 128
      BS(2) = FS(2) * (7.0*EMN - 1.0)/6.0
C  TEMPERATURE     EQ. 130
          R(3) = (7.0*EMN - 1.0)*(EMN + 5.0) / (36.0*EMN)
      BS(3) = FS(3) * R(3)
C  SPEED OF SOUND
      BS(4) = FS(4) * SQRT(R(3))
C  VISCOSITY
      IF(BS(3).GE.225.0) BS(5)=2.27*BS(3)**1.5/((BS(3)+198.6)*10.0**8)
      IF(BS(3).LT.225.0) BS(5)=0.80382436E-9 * BS(3)
C  MACH NUMBER     EQ. 132
      BS6SQ = (36.0*FS(6)**2*EMN - 5.0*(EMN - 1.0)*
     1        (7.0*EMN + 5.0)) / ((7.0*EMN - 1.0) * (EMN + 5.0))
```

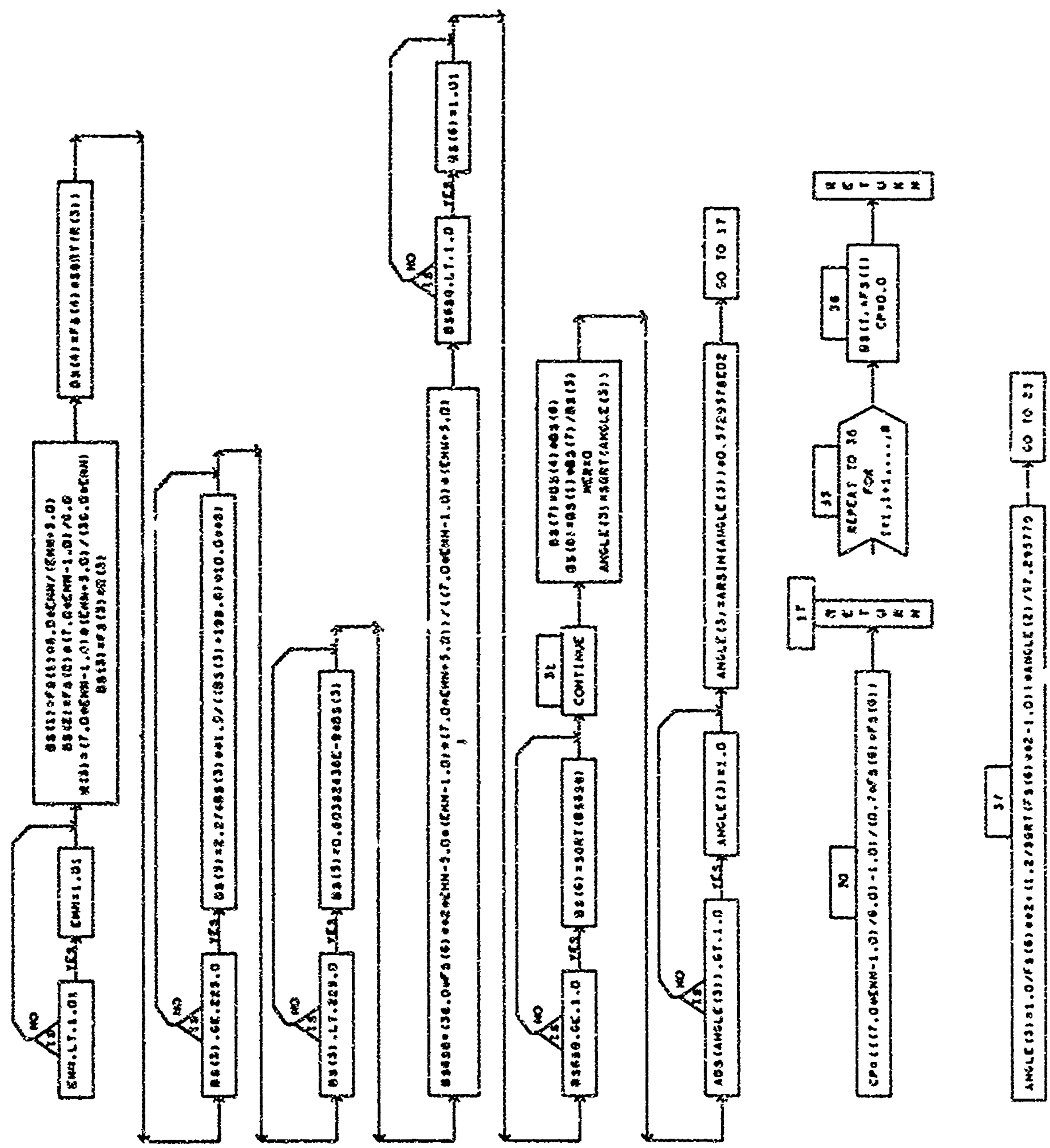
DECK AROM

```
      IF (BS6SQ .LT. 1.0) BS(6) = 1.01
      IF (BS6SQ .GE. 1.0) BS(6) = SQRT(BS6SQ)
   31 CONTINUE
C  VELOCITY
      BS(7) = BS(4) * BS(6)
C  REYNOLDS NUMBER PER FOOT
      BS(8) = BS(1) * BS(7)/BS(5)
      MER = 0
C  SHOCK ANGLE
      ANGLE(3) = SQRT(ANGLE(3))
      IF (ABS(ANGLE(3)) .GT. 1.0) ANGLE(3) = 1.0
      ANGLE(3) = ARSIN(ANGLE(3)) * 0.5729578E02
      GO TO 17
C
   30 CP = (((7.0*EMN - 1.0)/6.0) - 1.0) / (0.7*FS(6)*FS(6))
C
   17 RETURN
C
C     ANGLE(2) IS ZERO, SET BS(I) = FS(I), CP = 0.0, AND EXIT

   35 DO 36 I= 1,8
   36 BS(I)  = FS(I)
      CP     = 0.0
      RETURN
C  USE WEAK OBLIQUE SHOCK RELATIONSHIP (LIEPMAN AND ROSHKO, P.92)
   37 ANGLE(3) = 1.0/FS(6)**2 + (1.2/SQRT(FS(6)**2-1.0))*ANGLE(2)/
     1           57.295779
      GO TO 23
      END
```

SUBROUTINE COMPR




GO TO 14
R(2)-R(3)
IF
0
-
+
GO TO 8
7
L=1
GO TO 9
8
L=2
9
CONTINUE
GO TO 14
YES
IS
K.EQ.L
NO
GO TO 13
R(1)-R(3)
GO TO 12
11
COMPUTED GO TO
IF THE VALUE OF K IS 1 2 TRANSFER TO STATEMENT 15 13
12
COMPUTED GO TO
IF THE VALUE OF K IS 1 2 TRANSFER TO STATEMENT 13 15
13
ANGLE(3)=R(1)
GO TO 15
14
ANGLE(3)=R(2)
GO TO 16
15
ANGLE(3)=R(3)
16
CONTINUE
GO TO 20
ANGLE(3).GE.0.0
GO TO 19
IPRINT.NE.1
81
WRITE
WRITE(6,18)
19
ANGLE(3)=0.0
20
CONTINUE
GO TO 23
ANGLE(3).LE.1.0
GO TO 21
IPRINT.NE.1
81
WRITE
WRITE(6,22)
21
ANGLE(3)=1.0
23
ENN=FS(6)**2*ANGLE(3)
GO TO 30
ISDET.EQ.2
24
CONTINUE



SYMBOLS USED IN SUBROUTINE COMPR

ANGLE	R	A	FLOW ANGLE ARRAY	COMPR
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	COMPR
B	R	U	VARIABLE IN CUBIC EQUATION	COMPR
BS	R	C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	COMPR
BS6SQ	R	U	MACH NUMBER SQUARED	COMPR
C	R	U	VARIABLE IN CUBIC EQUATION	COMPR
CASE	I	C	CASE NUMBER	COMPR
CP	R	A	PRESSURE COEFFICIENT	COMPR
CPSTAG	R	A	MODIFIED NEWTONIAN CORRELATION FACTOR, K	COMPR
D	R	U	VARIABLE IN CUBIC EQUATION	COMPR
EMN	R	U	MACH NUMBER TIMES SINE THETA SQUARED	COMPR
ERROR	I	C	ERROR FLAG	COMPR
ETAC	R	U	PRANDTL-MEYER EXPANSION CORRECTION FACTOR	COMPR
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	COMPR
IFIRST	I	A	FIRST POINT FLAG FOR NEWTPM	COMPR
IM	I	C	ELEMENT ROW NUMBER ARRAY	COMPR
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	COMPR
IPRINT	I	A	PRINT FLAG	COMPR
ISDET	I	A	DATA GENERATION CONTROL FLAG	COMPR
ISE	I	U	DATA GENERATION CONTROL FLAG	COMPR
K	I	C	NUMBER OF ELEMENTS	COMPR
L	I	U	FLAG	COMPR
LS	I	C	NUMBER OF ELEMENTS	COMPR
MACHO	R	U	MACH NUMBER	COMPR
MACHSQ	R	U	MACH NUMBER SQUARED	COMPR
MACH1	R	U	MACH NUMBER	COMPR
MER	I	A	ERROR FLAG	COMPR
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY -X	COMPR
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	COMPR
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	COMPR
PAGE	I	C	PAGE NUMBER	COMPR
PFS	R	A	FREE-STREAM PRESSURE-LB/FT SQUARED	COMPR
R	R	D	ROOTS OF CUBIC EQUATION	COMPR
TFS	R	A	FREE-STREAM TEMPERATURE-DEGREE R	COMPR
TITLE	R	C	TITLE	COMPR
W	R	U	PARAMETER IN CUBIC EQUATION	COMPR

```
SYMBOLS USED IN SUBROUTINE COMPR

XCENT2   R  C  ELEMENT CENTROID ARRAY-X                                COMPR
Y        R  U  PARAMETER IN CUBIC EQUATION                             COMPR
YCENT2   R  C  ELEMENT CENTROID ARRAY-Y                                COMPR
Z        R  U  PARAMETER IN CUBIC EQUATION                             COMPR
ZCENT2   R  C  ELEMENT CENTROID ARRAY-Z                                COMPR
```

15. SUBROUTINE EXPAND (DECK ARON)

This routine calculates the local flow conditions on a surface by using conventional Prandtl-Meyer relationships (NACA TR 1135).

a. Algorithm

Calculate the Prandtl-Meyer angle after the expansion. Check if the flow is to be compressed to subsonic conditions or if it has expanded to an infinite Mach number (100). An iterative procedure is used to determine the flow conditions after the expansion fan.

b. Input/Output

None

c. Error

None

d. Subroutines Required

None

e. Argument List

(ANGLE, MER, IPRINT, ISDET, CP)

f. Length

2104 bytes

DECK ARON

```
      SUBROUTINE EXPAND (ANGLE,MER,IPRINT,ISDET,CP)
C
C  GIVEN THE FREE STREAM CONDITIONS (FS) AND THE  TURNING  ANGLE IN
C  DEGREES (ANGLE(2)), THIS SUBROUTINE PERFORMS AN ISENTROPIC PRANDTL-
C  MEYER EXPANSION(ANGLE(2).GT.0.) OR COMPRESSION(ANGLE(2).LT.0.)
C
      DIMENSION  FS(8),ANGLE(3),BS(8),A(2),C(2)
      DIMENSION  TITLE(15)
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),
     1  YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)
      COMMON  CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1  AREA2,IN,IM,K,LS,FS,BS
C
      REAL  NX2, NY2, NZ2
      INTEGER  CASE,PAGE,ERROR
      REAL  NU1, NU2, NU1D, NU2D
C
C  CHECK IF FREE STREAM MACH NO. .GE. 1.0
      IF (FS(6).GE.1.0) GO TO 1
C  INPUT MACH NO. SUBSONIC. FOR PROGRAM CONTINUITY SET = 1.0 AND GO ON
      EMSQ = 1.0
      GO TO 2
C
C  SQUARE FREE STREAM MACH NO.
    1 EMSQ = FS(6)**2
C   DEFINE GAMMA RATIO FUNCTION, GR = SQRT((G+1)/(G-1)). FOR G=1.4
    2 GR = SQRT(6.0)
C  CALCULATE PRANDTL-MEYER ANGLE FOR FREE STREAM CONDITIONS USING
C  EQUATION 171C OF TR 1135  (RADIANS)
      NU1 = GR*ATAN(SQRT (EMSQ-1.)/GR ) - ATAN(SQRT(EMSQ-1.))
C
C  CALCULATE PRANDTL-MEYER ANGLE AFTER THE EXPANSION (RADIANS)
      NU1D = NU1 * 57.295779
      NU2D = NU1D + ANGLE(2)
      NU2 = NU2D/57.295779
```

```
DECK ARON

C                                                                       ARON  0360
C  CHECK IF FLOW COMPRESSED TC SUBSONIC.                                ARON  0370
C                                                                       ARON  0380
      IF (NU2D.GT.0.) GO TO 21                                          ARON  0390
C  NU2D .LE. 0.0, RETURN SONIC CONDITIONS                               ARON  0400
C                                                                       ARON  0410
      BS(6) = 1.0                                                       ARON  0420
      MER = 2                                                           ARON  0430
      GO TO 13                                                          ARON  0440
C                                                                       ARON  0450
C  CHECK IF FLOW HAS EXPANDED TO AN INFINITE MACH NUMBER (TAKEN         ARON  0460
C  AS 100. FOR ALL PRACTICAL PURPOSES).  IF SO, RETURN ZERO PRESSURE.   ARON  0470
   21 IF (NU2D.LT. 127.6) GO TO 22                                      ARON  0480
      MER = 2                                                           ARON  0490
      BS(6) = 100.0                                                     ARON  0500
      GO TO 13                                                          ARON  0510
C                                                                       ARON  0520
C                                                                       ARON  0530
C  START OF ITERATION TO FIND MACH NC. DOWNSTREAM                       ARON  0540
C  SET INITIAL CONDITIONS AND TOLERANCE                                 ARON  0550
   22 I = 0                                                             ARON  0560
      A(2) = 0.0                                                        ARON  0570
      C(2) = 0.0                                                        ARON  0580
      EPS = 1.E-4                                                       ARON  0590
      JPATH = 1                                                         ARON  0600
C  JPATH CONTROLS THE LOGICAL PATH DURING THE ITERATION CYCLE.          ARON  0610
C  CALCULATE APPROXIMATE DOWNSTREAM MACH NO.                            ARON  0620
      BS(6) = FS(6)*(1. + (NU2-NU1)*(1.+0.2*EMSQ)/SQRT(EMSQ-1.))        ARON  0630
      IF (BS(6).GT.1.0) GO TO 3                                         ARON  0640
      BS(6) = 1.01                                                      ARON  0650
C  SET ITERATION COUNTER AND CHECK FOR MAXIMUM                          ARON  0660
    3 I    = I + 1                                                      ARON  0670
C                                                                       ARON  0680
C  CHECK NUMBER OF ITERATIONS COUNTER                                   ARON  0690
    6 IF (I .LE. 20) GO TO 9                                            ARON  0700
      WRITE (6,8) I                                                     ARON  0710
```

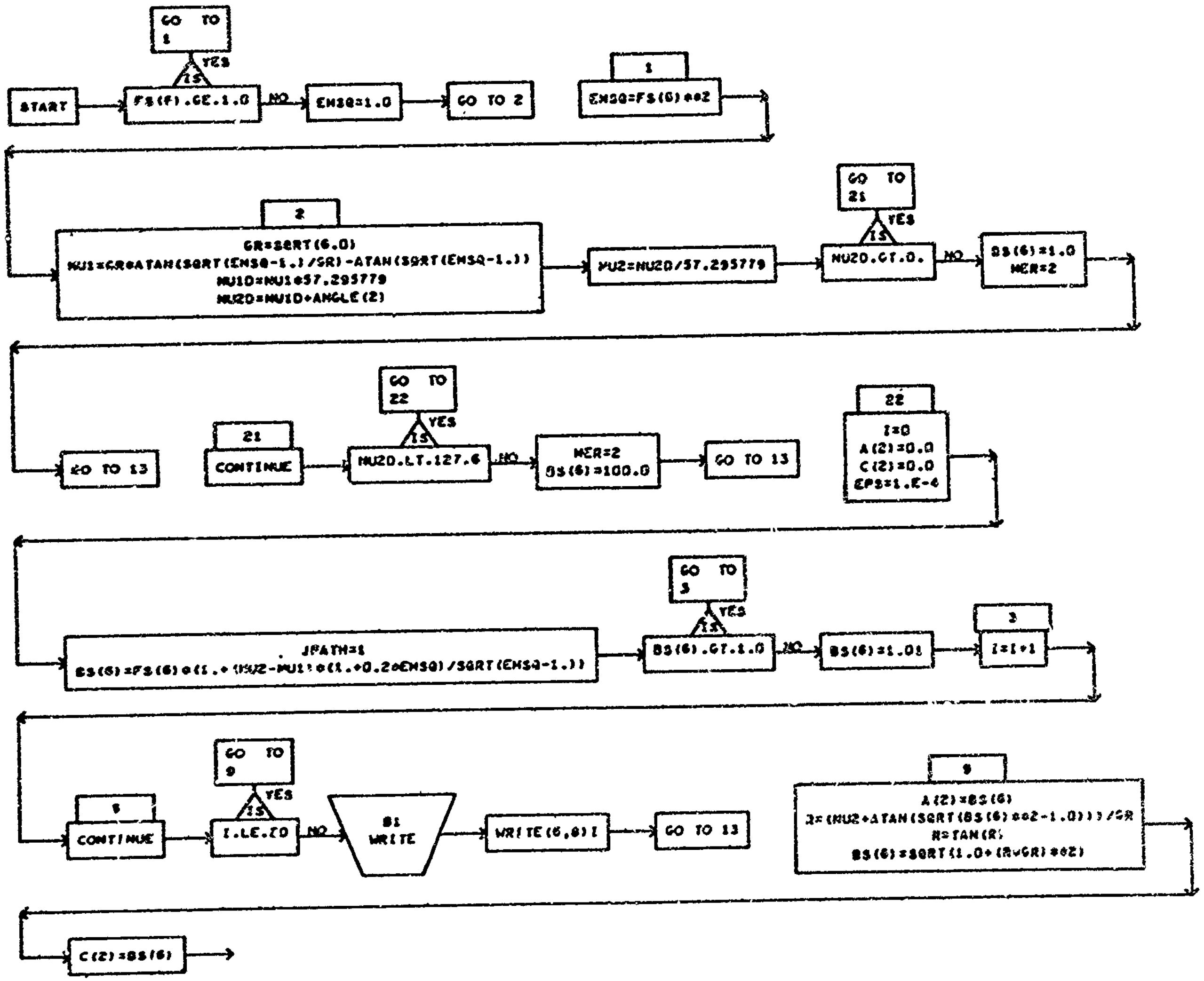
DECK ARON

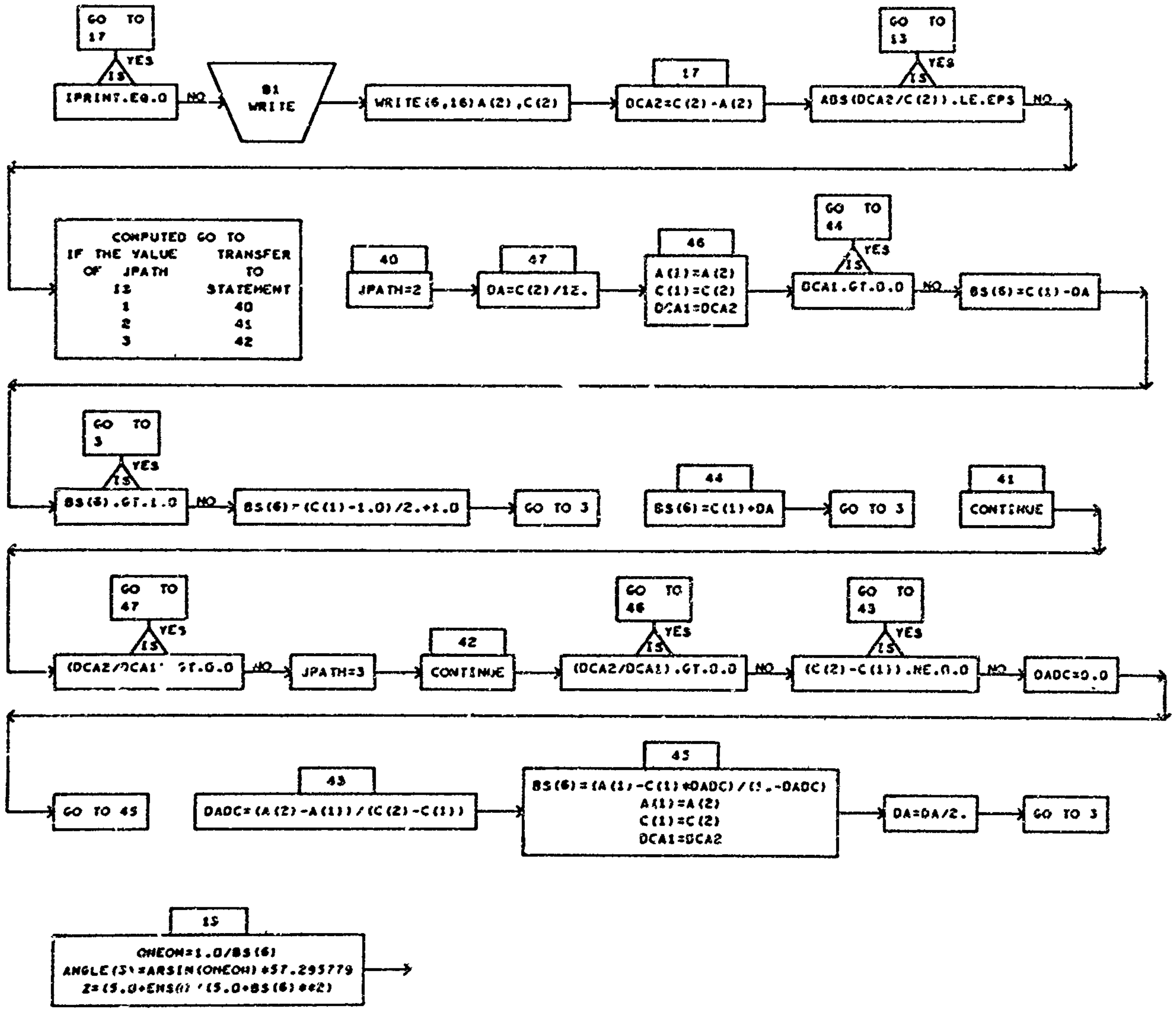
```
    8  FORMAT (1HO,I4,42H ITERATIONS IN EXPANSION ROUTINE. THE LAST
      1 25H VALUE HAS BEEN ACCEPTED. )
       GO TO 13
C
    9  A(2) = BS(6)
C
       R =  (NU2     + ATAN(SQRT(BS(6)**2 - 1.0)))/GR
       R = TAN(R)
       BS(6) = SQRT(1.0 + (R*GR)**2)
C
       C(2) = BS(6)
C
C  CHECK IF FLOW ITERATION IS TO BE PRINTED OUT
       IF (IPRINT .EQ. 0) GO TO 17
       WRITE (6,16) A(2), C(2)
   16   FORMAT (1H ,17X3HMA=F8.4,4X3HMC=F8.4 )
C
C  CHECK IF ITERATION ACCURACY HAS BEEN REACHED
   17  DCA2 = C(2)-A(2)
       IF (ABS(DCA2/C(2)).LE.EPS) GO TC 13
C
       GO TO (40,41,42), JPATH
   40  JPATH = 2
C  STEP ASSUMED VALUE BY AN ARBITRARY INCREMENT
C  EXPERIENCE HAS SHOWN THAT ONE-12TH OF C(2) TO BE A GOOD VALUE.
   47  DA = C(2)/12.
   46  A(1) = A(2)
       C(1) = C(2)
       DCA1 = DCA2
       IF (DCA1.GT.0.0) GO TO 44
       BS(6)= C(1) - DA
C  MAKE SURE THAT 2ND GUESS IS NOT CUT OF RANGE.
       IF(BS(6).GT.1.0) GO TO 3
       BS(6)= (C(1)-1.0)/2. + 1.0
       GO TO 3
   44  BS(6)= C(1) + DA
```

DECK ARON

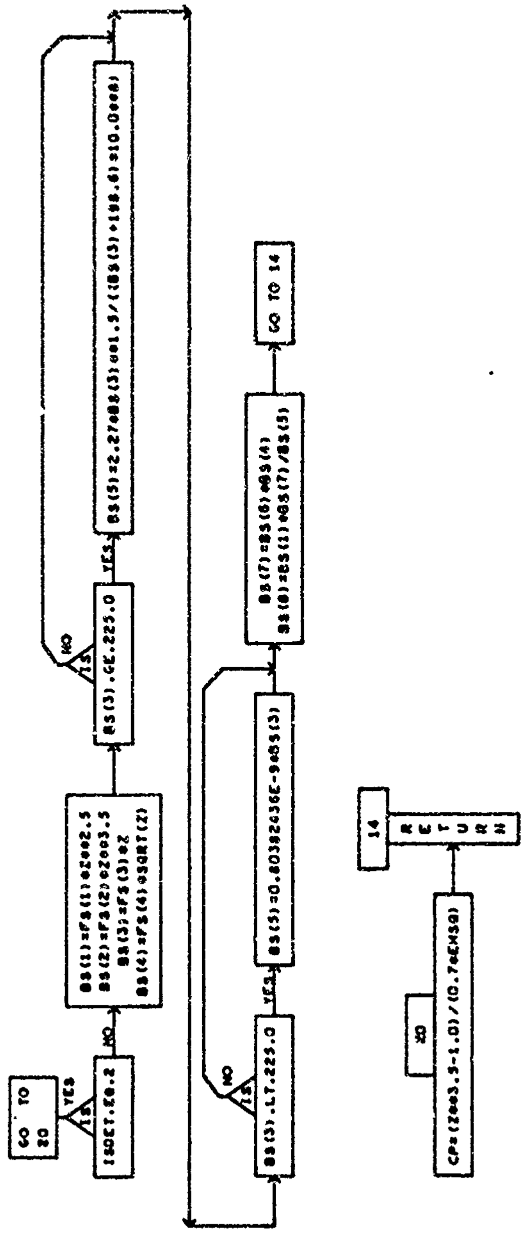
```
      GO TO 3
   41 IF ((DCA2/DCA1).GT.0.0) GO TO 47
      JPATH = 3
   42 IF ((DCA2/DCA1).GT.0.0) GO TO 46
C  CALCULATE MACH NUMBER AFTER EXPANSION USING 2 PREVIOUS ESTIMATES
      IF ((C(2)-C(1)) .NE. 0.0) GO TO 43
      DADC = 0.0
      GO TO 45
   43 DADC = (A(2)-A(1))/(C(2)-C(1))
   45 BS(6) = (A(1)-C(1)*DADC)/(1.-DADC)
      A(1) = A(2)
      C(1) = C(2)
      DCA1 = DCA2
      DA = DA/2.
      GO TO 3
C
C
C  CALCULATE FINAL CHARACTERISTICS BEHIND EXPANSION FAN
   13  ONEOM = 1.0 / BS(6)
      ANGLE(3) = ARSIN(ONEOM)*57.295779
      Z      = (5.0 + EMSQ)      / (5.0 + BS(6)**2)
      IF (ISDET .EQ. 2) GO TO 20
      BS(1) = FS(1) *Z**2.5
      BS(2) = FS(2) *Z**3.5
      BS(3) = FS(3) *Z
      BS(4) = FS(4) *SQRT(Z)
      IF(BS(3).GE.225.0) BS(5)=2.27*BS(3)**1.5/((BS(3)+198.6)*10.0**8)
      IF(BS(3).LT.225.0) BS(5)=0.80382436E-9 * BS(3)
      BS(7) = BS(6) * BS(4)
      BS(8) = BS(1) * BS(7)/BS(5)
      GO TO 14
C
   20 CP = (Z**3.5 - 1.0) / (0.7*EMSQ)

   14 RETURN
C
      END
```

SUBROUTINE EXPAND





GO TO 17
YES
IS
IPRINT.EQ.0
NO
B1 WRITE
WRITE(6,16)A(2),C(2)
17
DCA2=C(2)-A(2)
GO TO 13
YES
IS
ABS(DCA2/C(2)).LE.EPS
NO
COMPUTED GO TO
IF THE VALUE OF JPATH IS
TRANSFER TO STATEMENT
1 40
2 41
3 42
40
JPATH=2
47
DA=C(2)/12.
46
A(1)=A(2)
C(1)=C(2)
DCA1=DCA2
GO TO 44
YES
IS
DCA1.GT.0.0
NO
BS(6)=C(1)-DA
GO TO 3
YES
IS
BS(6).GT.1.0
NO
BS(6)=(C(1)-1.0)/2.+1.0
GO TO 3
44
BS(6)=C(1)+DA
GO TO 3
41
CONTINUE
GO TO 47
YES
IS
NO
JPATH=3
42
CONTINUE
GO TO 46
YES
IS
(DCA2/DCA1).GT.0.0
NO
GO TO 43
YES
IS
(C(2)-C(1)).NE.0.0
NO
DADC=0.0
GO TO 45
43
DADC=(A(2)-A(1))/(C(2)-C(1))
45
A(1)=A(2)
C(1)=C(2)
DCA1=DCA2
DA=DA/2.
GO TO 3
15
ONEOM=1.0/BS(6)
ANGLE(3)=ARSIN(ONEOM)*57.295779


ISECT.EQ.2
IS
YES
GO TO 20
NO
BS(1)=FS(1)*Z**2.5
BS(2)=FS(2)*Z**3.5
BS(3)=FS(3)*Z
BS(4)=FS(4)*SQRT(Z)
BS(3).GE.225.0
IS
NO
YES
BS(5)=2.27*BS(3)**1.5/((BS(3)+198.6)*10.0**8)
BS(3).LT.225.0
IS
NO
YES
BS(5)=0.80382436E-9*BS(3)
BS(7)=BS(6)*BS(4)
BS(8)=BS(1)*BS(7)/BS(5)
GO TO 14
20
CP=(Z**3.5-1.0)/(0.7*EMSQ)
14
RETURN

SYMBOLS USED IN SUBROUTINE EXPAND

A	R	D	ITERATION VARIABLE ARRAY	EXPAND
ANGLE	R	A	FLOW ANGLE ARRAY	EXPAND
AREA2	R	C	QNADRILATERAL ELEMENT AREA ARRAY	EXPAND
BS	R	C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	EXPAND
C	R	D	ITERATION VARIABLE ARRAY	EXPAND
CASE	I	C	CASE NUMBER	EXPAND
CP	R	A	PRESSURE COEFFICIENT	EXPAND
DA	R	U	ITERATION INCREMENT	EXPAND
DADC	R	U	EXPANSION ITERATION PARAMETER	EXPAND
DCA1	R	U	EXPANSION ITERATION PARAMETER	EXPAND
DCA2	R	U	EXPANSION ITERATION PARAMETER	EXPAND
EMSQ	R	U	MACH NUMBER SQUARED	EXPAND
EPS	R	U	ITERATION ACCURACY PARAMETER	EXPAND
ERROR	I	C	ERROR FLAG	EXPAND
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	EXPAND
GR	R	U	GAMMA RATIO FUNCTION	EXPAND
I	I	U	ITERATION COUNTER	EXPAND
IM	I	C	ELEMENT ROW NUMBER ARRAY	EXPAND
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	EXPAND
IPRINT	I	A	PRINT FLAG	EXPAND
ISDET	I	A	DATA GENERATION CONTROL FLAG	EXPAND
JPATH	I	U	CONTROL FLAG FOR ITERATION PATH	EXPAND
K	I	C	NUMBER OF ELEMENTS	EXPAND
LS	I	C	NUMBER OF ELEMENTS	EXPAND
MER	I	A	ERROR FLAG	EXPAND
NU1	R	U	INITIAL PRANDTL-MEYER ANGLE, RADIANS	EXPAND
NU1D	R	U	INITIAL PRANDTL-MEYER ANGLE, DEGREES	EXPAND
NU2	R	U	FINAL PRANDTL-MEYER ANGLE, RADIANS	EXPAND
NU2D	R	U	FINAL PRANDTL-MEYER ANGLE, DEGREES	EXPAND
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	EXPAND
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	EXPAND
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	EXPAND
ONEOM	R	U	SINE OF ANGLE 3	EXPAND
PAGE	I	C	PAGE NUMBER	EXPAND
R	R	U	VARIABLE IN MACH NUMBER EQUATION	EXPAND
TITLE	R	C	TITLE	EXPAND

```
SYMBOLS USED IN SUBROUTINE EXPAND

XCENT2   R   C   ELEMENT CENTROID ARRAY-X                         EXPAND
YCENT2   R   C   ELEMENT CENTROID ARRAY-Y                         EXPAND
Z        R   U   FLOW CHARACTERISTIC PARANETERS                   EXPAND
ZCENT2   R   C   ELEMENT CENTROID ARRAY-Z                         EXPAND
```

16. SUBROUTINE NEWTPM (DECK AROO)

This subroutine calculates the pressure coefficients on a surface by the blunt-body Newtonian + Prandtl-Meyer method.

a. Algorithm

This first section of this routine performs an iteration to find the matching point Mach number. A Prandtl-Meyer expansion is then calculated from the matching point condition to the local element slope using the EXPAND routine. Finally, the pressure coefficient and the local flow properties are calculated.

b. Input/Output

None

c. Error

None

d. Subroutines Required

EXPAND

e. Argument List

(ANGLE, EMN, CP, ETAC, IPRINT, MER, CPSTAG, TFS, PFS, ISE, IFIRST)

f. Length

2862 bytes

DECK AROD

```
      SUBROUTINE NEWTPM (ANGLE,EMN,CP,ETAC,IPRINT,MER,CPSTAG,
     1  TFS,PFS,ISE,IFIRST )
C
C  THIS SUBROUTINE CALCULATES THE SURFACE CONDITIONS USING THE BLUNT
C  BODY SHOCK-EXPANSION TECHNIQUE OF KAUFMAN, JOURNAL OF THE
C  ASTRONAUTICAL SCIENCES, VOL X, NO.2 SUMMER 1963.
C
      DIMENSION FS(8),BS(8),ANGLE(3)
      DIMENSION  TITLE(15)
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),
     1  YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)
      COMMON  CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1  AREA2,IN,IM,K,LS,FS,BS
      REAL  NX2, NY2, NZ2
      INTEGER  CASE,PAGE,ERROR
C
      REAL MACHO, MACH, MSUBQ, MU, MACHSQ, M1, M2
C
C
      IF (CPSTAG .LE. 0.01) CPSTAG = 2.0
      IF (ETAC .LE. 0.01) ETAC = 1.0
      MER = 0
      IF (IFIRST .EQ. 1) GO TO 11
      IFIRST = 1
      MACHO = FS(6)
C
C
C  THE FOLLOWING SECTION PERFORMS AN ITERATION TO FIND THE MATCHING
C  POINT MACH NUMBER. IT ONLY WORKS FOR GAMMA = 1.4.
C
C  SET GAMMA = G = 1.4
      G = 1.4
      J = 0
C  SOLVE FOR PRESSURE RATIO  (1.0/ EQ. 100 OF TR 1135)
      PCAP= (2./((G+1.0)*MACHO*MACHO))**(G/(G-1.0))  *
```

DECK AROO

```
   1    ((2.*G*MACHO*MACHO-(G-1.0))/(G+1.0))**(1.0/(G-1.0))
C
C  ASSUME MACH SUB Q = 1.35
      MSUBQ = 1.35
C
C  CALCULATE  Q  (SEE KAUFMAN)
   9  Q =  (2.0/(2.0+ (G-1.0)*MSUBQ*MSUBQ))**(G/(G-1.0))
C
C  CALCULATE P SUB C  (EQ 9 OF KAUFMAN)
      PC = Q * (1.0 - (G*G*MSUBQ**4*Q)/ (4.0*(MSUBQ*MSUBQ-1.0)*(1.-Q)))
      CPQ = (2.0 / (G*MACHO*MACHO)) * (Q/PC - 1.0)
      IF (IPRINT .EQ. 0) GO TO 12
      WRITE (6,10) MSUBQ,PC,CPQ
  10  FORMAT (1H ,17X7HMSUBQ =F9.6,6H  PC =1PE11.4,7H  CPQ =E11.4 )
C
C  CHECK ITERATION ACCURACY
  12  IF (ABS(PCAP-PC) .LT. 0.0000001) GO TO 7
C
C  SET UP ITERATION TERMS
      P1 = P2
      P2 = PC
      M1 = M2
      M2 = MSUP
C
C  STEP ITERATION COUNTER AND CHECK CYCLE
      J = J + 1
      IF (J .GT. 1) GO TO 8
       MSUBQ= 1.7
       GO TO 9
   8  IF (J.GT. 20) GO TO 7
C  ESTIMATE NEW M
      MSUBQ = M1 + (PCAP-P1)*(M2-M1)/(P2-P1)
C  CHECK NEW ESTIMATE FOR M
      IF (MSUBQ .GT. 1.75) MSUBQ = 1.75
      IF (MSUBQ .LT. 1.35) MSUBQ = 1.35
      IF (MSUBQ.EQ.M1 .OR. MSUBQ.EQ.M2) MSUBQ = MSUBQ + 0.0001
```

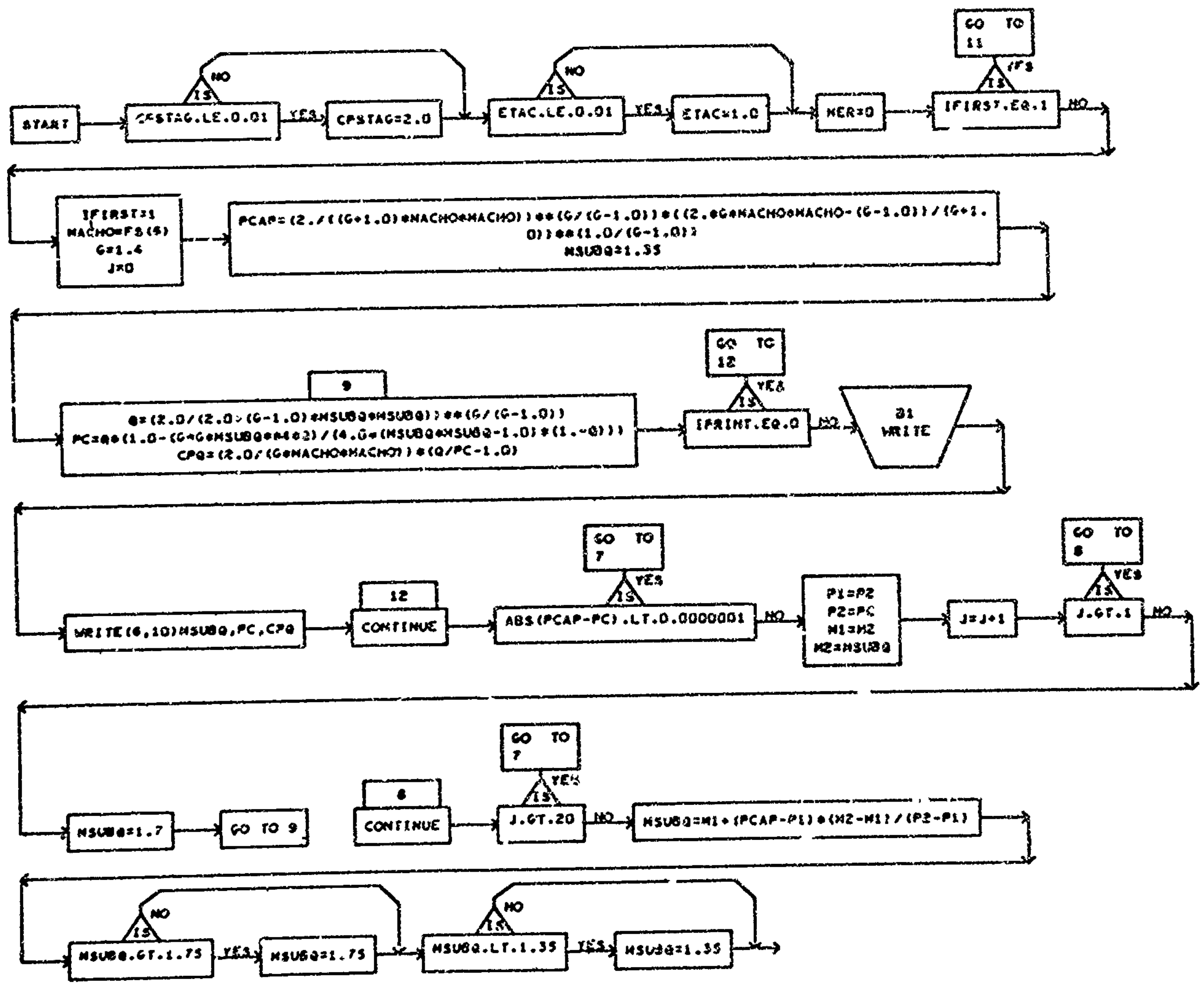
DECK AROO

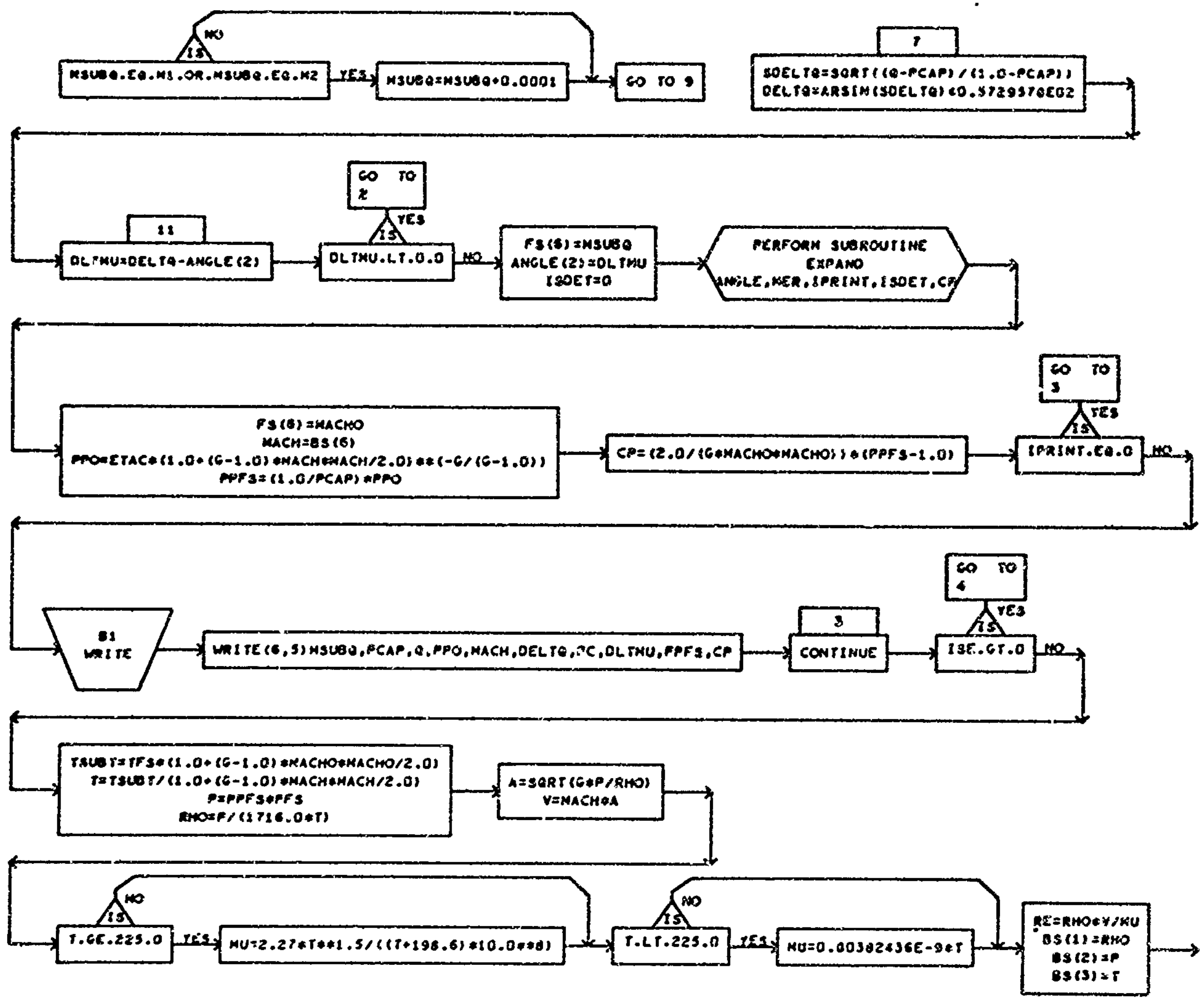
```
      GO TO 9
C
C  CALCULATE MATCHING POINT IMPACT ANGLE
 7    SDELTQ = SQRT((Q-PCAP)/(1.0-PCAP))
      DELTQ = ARSIN(SDELTQ) * 0.5729578E02
C
C  CALCULATE EXPANSION ANGLE FROM MATCHING POINT
 11   DLTMU = DELTQ - ANGLE(2)
C
C  CHECK IF FLOW WILL EXPAND AT LEAST TO MATCHING POINT
      IF (DLTMU .LT. 0.0) GO TO 2
C
C  DETERMINE MACH NUMBER ON SURFACE
      FS(6) = MSUBQ
      ANGLE(2) = DLTMU
C
      ISDET = 0
      CALL EXPAND (ANGLE,MER,IPRINT,ISDET,CP)
      FS(6) = MACHO
C  SET UP SURFACE MACH NUMBER
      MACH = BS(6)
C
C  CALCULATE SURFACE PRESSURE RATIO (EQ. 44 OF TR 1135)
      PPO = ETAC * (1.0 + (G-1.0)*MACH*MACH/2.0)**(-G/(G-1.0))
C
C  CALCULATE  P / P FREE STREAM
      PPFS = (1.0/PCAP)*PPO
C  CALCULATE PRESSURE COEFFICIENT ON SURFACE
      CP =  (2.0/(G*MACHO*MACHO))* (PPFS - 1.0)
      IF (IPRINT .EQ. 0) GO TO 3
      WRITE (6,5) MSUBQ,PCAP,Q,PPO,MACH,DELTQ,PC,DLTMU,PPFS,CP
 5    FORMAT (1H ,17X23HSHOCK-EXPANSION  M Q  =F7.5,8H  P CAP=1PE11.4,
     1  8H  Q     =E11.4,8H  P/PO =E11.4,7H  MACH=0PF7.3,/1H ,17X
     2 23HCALCULATIONS      DELTQ=F7.3,8H  P C  =1PE11.4, 8H  DLTMU=
     3  E11.4,8H  P/PFS=E11.4,5H  CP=0PF9.5 )
C  CHECK IF FLOW CONDITIONS ARE NEEDED
```

```
DECK AROO

   3  IF (ISE .GT. 0) GO TO 4
C
C  CALCULATE FREE STREAM TOTAL TEMPERATURE (EQ. 43 OF TR 1135)
      TSUBT = TFS * (1.0 +  (G-1.0)*MACHO*MACHO/2.0)
C
C  CALCULATE TEMPERATURE AFTER EXPANSION (IN RANKINE)
      T  = TSUBT /  (1.0 + (G-1.0)*MACH*MACH / 2.0)
C
C  CALCULATE SURFACE PRESSURE
      P  = PPFS * PFS
C
C  CALCULATE DENSITY (EQ. 26 OF TR 1135)
      RHO  = P / (1716.0*T)
C
C  CALCULATE LOCAL SPEED OF SOUND
      A =  SQRT(G*P/RHO)
C
C  CALCULATE LOCAL VELOCITY
      V = MACH * A
C
C  CALCULATE VISCOSITY
      IF(T.GE.225.0) MU = 2.27*T**1.5/((T+198.6)*10.0**8)
      IF(T.LT.225.0) MU = 0.80382436E-9 * T
C
C  CALCULATE REYNOLDS NUMBER PER FOOT (EQ) B1 TR1135)
      RE  = RHO * V / MU
C
C  SET UP DATA FOR USE BACK IN OTHER SUBROUTINES
      BS(1) = RHO
      BS(2) = P
      BS(3) = T
      BS(4) = A
      BS(5) = MU
      BS(6) = MACH
      BS(7) = V
      BS(8) = RE
```

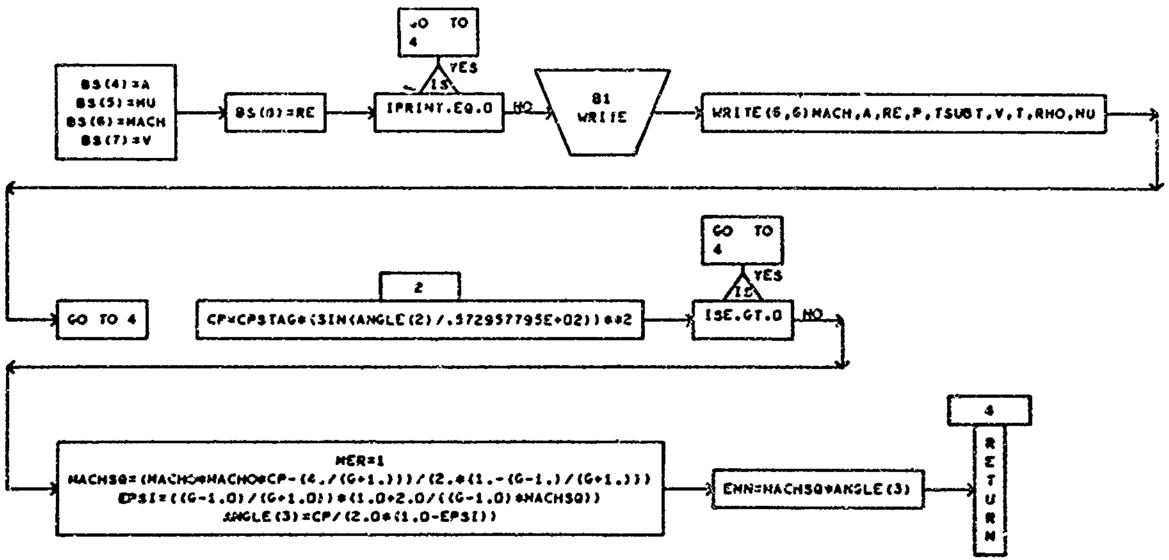
```
DECK AROO

      IF (IPRINT .EQ. 0) GO TO 4                                        AROO 1440
      WRITE (6,6) MACH,A,RE,P,TSUBT,V,T,RHO,MU                          AROO 1450
    6 FORMAT (1H ,34X6HMACH =F7.5,8H  A    =1PE11.4,8H  RE   =E11.4,    AROO 1460
     1 8H  P    =E11.4,7H  TTOT=0PF7.1,/1H ,34X6HV    =F7.1,8H  T    =  AROO 1470
     2 1PE11.4,8H  RHO  =E11.4,8H  MU   =E11.4  )                       AROO 1480
C                                                                       AROO 1490
C                                                                       AROO 1500
      GO TO 4                                                           AROO 1510
C                                                                       AROO 1520
C                                                                       AROO 1530
C   FLOW HAS NOT REACHED THE MATCHING POINT. USE NEWTONIAN CALCULATIONS AROO 1540
C   AND SHOCK DETACHED METHOD SUGGESTED BY SMYTH.                       AROO 1550
    2 CP = CPSTAG * (SIN(ANGLE(2)/.572957795E+02))**2                   AROO 1560
C                                                                       AROO 1570
C   CHECK IF FLOW CONDITIONS ARE NEEDED                                 AROO 1580
      IF (ISE .GT. 0) GO TO 4                                           AROO 1590
      MER = 1                                                           AROO 1600
C                                                                       AROO 1610
C   CALCULATE FLOW DATA FOR DETACHED CONDITIONS                         AROO 1620
C   CALCULATE SQUARE OF MACH NUMBER NORMAL TO EFFECTIVE SHOCK           AROO 1630
      MACHSQ=(MACHO*MACHO*CP-(4./(G+1.)))/(2.*(1.-(G-1.)/(G+1.)))       AROO 1640
C   CALCULATE EFFECTIVE DENSITY RATIO                                   AROO 1650
      EPSI = ((G-1.0)/(G+1.0))* (1.0 + 2.0/((G-1.0)*MACHSQ))            AROO 1660
C   CALCULATE THE EFFECTIVE SHOCK ANGLE SQUARED                         AROO 1670
      ANGLE(3) = CP / (2.0*(1.0-EPSI))                                  AROO 1680
C   CALCULATE NORMAL MACH SQUARED TIMES SQUARE OF SHOCK ANGLE           AROO 1690
      EMN = MACHSQ * ANGLE(3)                                           AROO 1700
C                                                                       AROO 1710
    4 RETURN                                                            AROO 1720
C                                                                       AROO 1730
      END                                                               AROO 1740
```

SUBROUTINE NEWTPN





MSUBQ.EQ.M1.OR.MSUBQ.EQ.M2
IS
NO
YES
MSUBQ=MSUBQ+0.0001
GO TO 9
7
SDELTQ=SQRT((Q-PCAP)/(1.0-PCAP))
DELTQ=ARSIN(SDELTQ)*0.5729578E02
11
DLTMU=DELTQ-ANGLE(2)
DLTMU.LT.0.0
IS
YES
GO TO 2
NO
FS(6)=MSUBQ
ANGLE(2)=DLTMU
ISDET=0
PERFORM SUBROUTINE
EXPANO
ANGLE,MER,IPRINT,ISDET,CP
FS(8)=MACHO
MACH=BS(6)
PPO=ETAC*(1.0+(G-1.0)*MACH*MACH/2.0)**(-G/(G-1.0))
PPFS=(1.0/PCAP)*PPO
CP=(2.0/(G*MACHO*MACHO))*(PPFS-1.0)
IPRINT.EQ.0
IS
YES
GO TO 3
NO
81
WRITE
WRITE(6,5)MSUBQ,PCAP,Q,PPO,MACH,DELTQ,PC,DLTMU,PPFS,CP
3
CONTINUE
ISE.GT.0
IS
YES
GO TO 4
NO
TSUBT=TFS*(1.0+(G-1.0)*MACHO*MACHO/2.0)
T=TSUBT/(1.0+(G-1.0)*MACH*MACH/2.0)
P=PPFS*PFS
RHO=P/(1716.0*T)
A=SQRT(G*P/RHO)
V=MACH*A
T.GE.225.0
IS
NO
YES
MU=2.27*T**1.5/((T+198.6)*10.0**8)
T.LT.225.0
IS
NO
YES
MU=0.00382436E-9*T
RE=RHO*V/MU
BS(1)=RHO
BS(2)=P
BS(3)=T


BS(4)=A
BS(5)=NU
BS(6)=MACH
BS(7)=V
BS(8)=RE
IPRINT.EQ.0
IS
YES
GO TO 4
NO
81
WRITE
WRITE(6,6)MACH,A,RE,P,TSUBT,V,T,RHO,NU
GO TO 4
2
CP=CPSTAG*(SIN(ANGLE(2)/.572957795E+02))**2
ISE.GT.0
IS
YES
GO TO 4
NO
NER=1
MACHSQ=(MACH*MACH*CP-(4./(G+1.)))/(2.*(1.-(G-1.)/(G+1.)))
EPSI=((G-1.0)/(G+1.0))*(1.0+2.0/((G-1.0)*MACHSQ))
ANGLE(3)=CP/(2.0*(1.0-EPSI))
ENN=MACHSQ*ANGLE(3)
4
RETURN

SYMBOLS USED IN SUBROUTINE NEWTPM

A	R	U	SPEED OF SOUND	NEWTPM
ANGLE	R	A	FLOW ANGLE ARRAY	NEWTPM
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	NEWTPM
BS	R	C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	NEWTPM
CASE	I	C	CASE NUMBER	NEWTPM
CP	R	A	PRESSURE COEFFICIENT	NEWTPM
CPQ	R	U	PRESSURE COEFFICIENT AT MATCHING POINT	NEWTPM
CPSTAG	R	A	MODIFIED NEWTONIAN CORRELATION FACTOR, K	NEWTPM
DELTQ	R	U	IMPACT ANGLE AT MATCHING POINT	NEWTPM
DLTMU	R	U	EXPANSION ANGLE FROM MATCHING MOMENT	NEWTPM
EMN	R	A	MACH NUMBER TIMES SHOCK ANGLE SQUARED	NEWTPM
EPSI	R	U	EFFECTIVE DENSITY RATIO	NEWTPM
ERROR	I	C	ERROR FLAG	NEWTPM
ETAC	R	A	PRANDTL-MEYER EXPANSION CORRECTION FACTOR	NEWTPM
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	NEWTPM
G	R	U	RATIO OF SPECIFIC HEATS = 1.4	NEWTPM
IFIRST	I	A	FIRST POINT FLAG FOR USE IN NEWTPM	NEWTPM
IM	I	C	ELEMENT ROW NUMBER ARRAY	NEWTPM
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	NEWTPM
IPRINT	I	A	PRINT FLAG	NEWTPM
ISDET	I	U	DATA GENERATION CONTROL FLAG	NEWTPM
ISE	I	A	DATA GENERATION CONTROL FLAG	NEWTPM
J	I	U	ITERATION COUNTER	NEWTPM
K	I	C	NUMBER OF ELEMENTS	NEWTPM
LS	I	C	NUMBER OF ELEMENTS	NEWTPM
MACH	R	U	MACH NUMBER	NEWTPM
MACHO	R	U	INITIAL MACH NUMBER	NEWTPM
MACHSQ	R	U	SQUARE OF MACH NUMBER NORMAL TO EFFECIVE SHOCK	NEWTPM
MER	I	A	ERROR FLAG	NEWTPM
MSUBQ	R	U	MACH NUMBER AT MATCHING POINT	NEWTPM
MU	R	U	VISCOSITY	NEWTPM
M1	R	U	FIRST ITERATION MACH NUMBER	NEWTPM
M2	R	U	SECOND ITERATION MACH NUMBER	NEWTPM
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	NEWTPM
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	NEWTPM
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	NEWTPM

SYMBOLS USED IN SUBROUTINE NEWTPM

P	R	U	PRESSURE	NEWTPM
PAGE	I	C	PAGE NUMBER	NEWTPM
PC	R	U	FREE-STREAM STATIC TO STAGNATION PRESSURE RATIO	NEWTPM
PCAP	R	U	FREE-STREAM STATIC TO STAGNATION PRESSURE RATIO	NEWTPM
PFS	R	A	FREE-STREAM PRESSURE, LBS / SQUARE FOOT	NEWTPM
PPFS	R	U	LOCAL TO FREE-STREAM PRESSURE RATIO	NEWTPM
PPO	R	U	SURFACE PRESSURE RATIO	NEWTPM
P1	R	U	FIRST ITERATION PRESSURE	NEWTPM
P2	R	U	SECOND ITERATION PRESSURE	NEWTPM
Q	R	U	MATCHING POINT TO FREE-STREAM STATIC PRESSURE RATIO	NEWTPM
RE	R	U	REYNOLDS NUMBER	NEWTPM
RHO	R	U	DENSITY	NEWTPM
SDELTQ	R	U	SINE OF MATCHING POINT IMPACT ANGLE	NEWTPM
T	R	U	TEMPERATURE	NEWTPM
TFS	R	A	FREE STREAM TEMPERATURE	NEWTPM
TITLE	R	C	TITLE	NEWTPM
TSUBT	R	U	FREE STREAM TOTAL TEMPERATURE	NEWTPM
V	R	U	VELOCITY	NEWTPM
XCENT2	R	C	ELEMENT CENTROID COORDINATE-X	NEWTPM
YCENT2	R	C	ELEMENT CENTROID COORDINATE-Y	NEWTPM
ZCENT2	R	C	ELEMENT CENTROID COORDINATE-Z	NEWTPM

```
SYMBOLS USED IN SUBROUTINE NEWTPM

P         R  U  PRESSURE                                                      NEWTPM
PAGE      I  C  PAGE NUMBER                                                   NEWTPM
PC        R  U  FREE-STREAM STATIC TO STAGNATION PRESSURE RATIO               NEWTPM
PCAP      R  U  FREE-STREAM STATIC TO STAGNATION PRESSURE RATIO               NEWTPM
PFS       R  A  FREE-STREAM PRESSURE, LBS / SQUARE FOOT                       NEWTPM
PPFS      R  U  LOCAL TO FREE-STREAM PRESSURE RATIO                           NEWTPM
PPO       R  U  SURFACE PRESSURE RATIO                                        NEWTPM
P1        R  U  FIRST ITERATION PRESSURE                                      NEWTPM
P2        R  U  SECOND ITERATION PRESSURE                                     NEWTPM
Q         R  U  MATCHING POINT TO FREE-STREAM STATIC PRESSURE RATIO           NEWTPM
RE        R  U  REYNOLDS NUMBER                                               NEWTPM
RHO       R  U  DENSITY                                                       NEWTPM
SDELTQ    R  U  SINE OF MATCHING POINT IMPACT ANGLE                           NEWTPM
T         R  U  TEMPERATURE                                                   NEWTPM
TFS       R  A  FREE STREAM TEMPERATURE                                       NEWTPM
TITLE     R  C  TITLE                                                         NEWTPM
TSUBT     R  U  FREE STREAM TOTAL TEMPERATURE                                 NEWTPM
V         R  U  VELOCITY                                                      NEWTPM
XCENT2    R  C  ELEMENT CENTROID COORDINATE-X                                 NEWTPM
YCENT2    R  C  ELEMENT CENTROID COORDINATE-Y                                 NEWTPM
ZCENT2    R  C  ELEMENT CENTROID COORDINATE-Z                                 NEWTPM
```

17. SUBROUTINE CONE (DECK AROP)

This subroutine solves for the local properties about a cone in supersonic flow using empirically derived equations.

a. Algorithm

Calculates the shock normal Mach number, surface pressure coefficient and, at the user's option the following local flow properties on the cone surface: pressure, density, temperature, velocity, speed of sound, Mach number, viscosity, and Reynolds number per foot. Solutions are empirically derived for a calorically perfect gas with ratio of specific heats equal to 1.40.

b. Input/Output

None

c. Error

None

d. Subroutines Required

None

e. Argument List

(ANGLE, CP, ISDT)

f. Length

1256 bytes

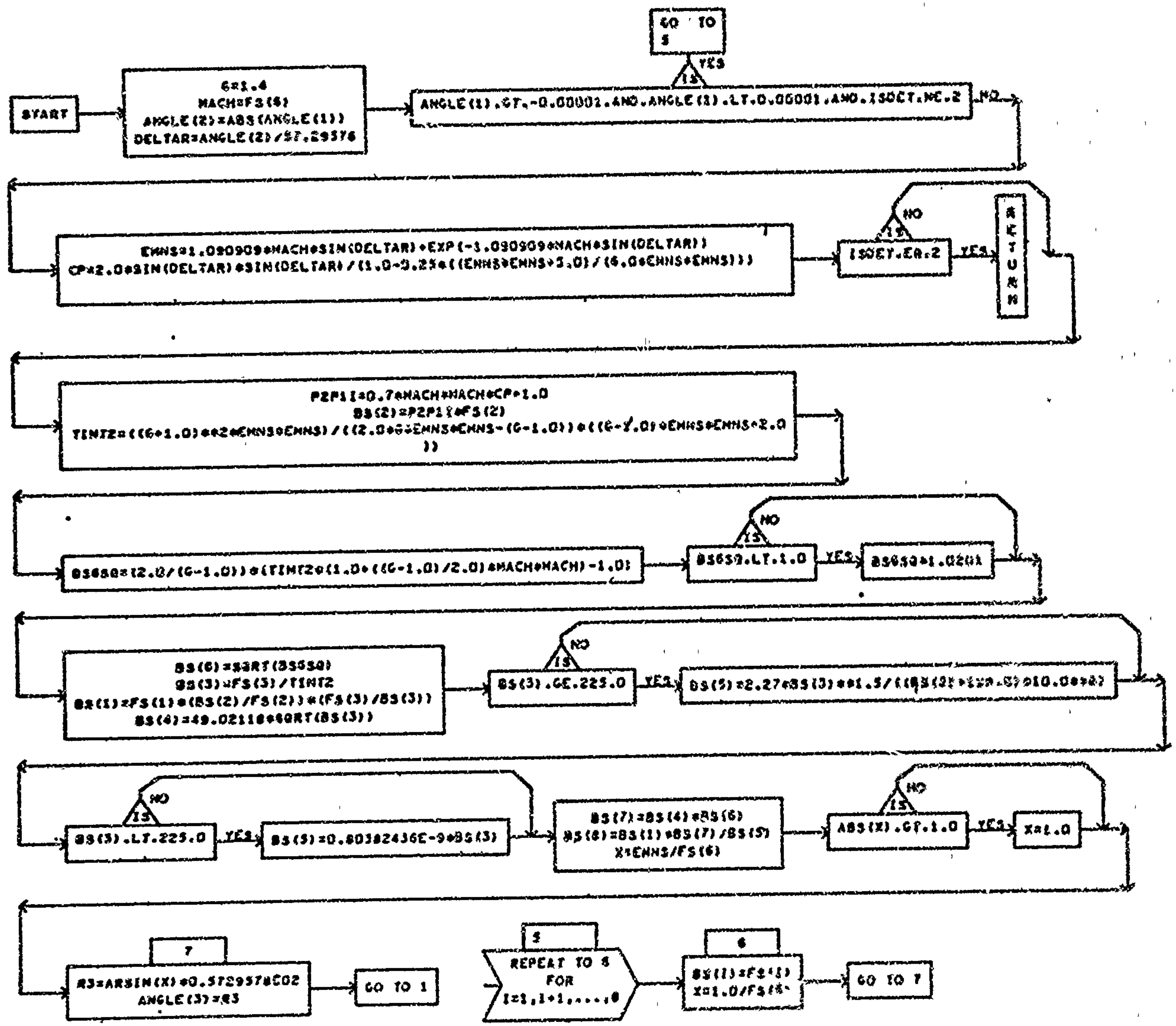
CONE

DECK AROP

```
      SUBROUTINE CONE (ANGLE,CP,ISDET)
C  THIS ROUTINE SOLVES FOR THE FLOW PROPERTIES ABOUT A CONE
C  USING EMPIRICAL EQUATIONS.
      DIMENSION ANGLE(3),FS(8),BS(8)
      DIMENSION  TITLE(15)
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),
     1  YCENT2( 300),ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)
      COMMON  CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1  AREA2,IN,IM,K,LS,FS,BS
      REAL  NX2,NY2,NZ2,MACH
      INTEGER  CASE,PAGE,ERROR
      G = 1.4
      MACH = FS(6)
      ANGLE(2) =  ABS(ANGLE(1))
      DELTAR = ANGLE(2)/57.29578
      IF (ANGLE(1).GT.-0.00001.AND.ANGLE(1).LT.0.00001.AND.ISDET.NE.2)
     1  GO TO 5
      EMNS = 1.090909*MACH*SIN(DELTAR) + EXP(-1.090909*MACH*SIN(DELTAR))
      CP = 2.0*SIN(DELTAR)*SIN(DELTAR)/(1.0-0.25*((EMNS*EMNS+5.0)/
     1      (6.0*EMNS*EMNS)))
      IF (ISDET .EQ. 2) RETURN
      P2P1I = 0.7*MACH*MACH*CP + 1.0
      BS(2) = P2P1I * FS(2)
      TINT2 = ((G+1.0)**2 *EMNS*EMNS)/((2.0*G*EMNS*EMNS-(G-1.0))*
     1         ((G-1.0)*EMNS*EMNS + 2.0))
      BS6SQ = (2.0/(G-1.0))*(TINT2*(1.0+((G-1.0)/2.0)*MACH*MACH)-1.0)
      IF (BS6SQ .LT. 1.0) BS6SQ = 1.0201
      BS(6) = SQRT(BS6SQ)
      BS(3) = FS(3) / TINT2
      BS(1) = FS(1)*(BS(2)/FS(2))*(FS(3)/BS(3))
      BS(4) = 49.02118 * SQRT(BS(3))
      IF(BS(3).GE.225.0) BS(5)=2.27*BS(3)**1.5/((BS(3)+198.6)*10.0**8)
      IF(BS(3).LT.225.0) BS(5)=0.80382436E-9 * BS(3)
      BS(7) = BS(4)*BS(6)
      BS(8) = BS(1)*BS(7)/BS(5)
```

```
DECK AROP

C     CALCULATE SHOCK ANGLE                                AROP  0360
      X = EMNS / FS(6)                                      AROP  0370
      IF (ABS(X) .GT. 1.0) X = 1.0                          AROP  0380
    7 R3 = ARSIN(X) * 0.5729578E02                          AROP  0390
      ANGLE(3) = R3                                         AROP  0400
      GO TO 1                                               AROP  0410
    5 DO 6 I=1,8                                            AROP  0420
    6   BS(I) = FS(I)                                       AROP  0430
        X = 1.0 / FS(6)                                     AROP  0440
        GO TO 7                                             AROP  0450
    1 RETURN                                                AROP  0460
      END                                                   AROP  0470
```

SUBROUTINE CONE



SYMBOLS USED IN SUBROUTINE CONE

ANGLE	R	A	FLOW ANGLE ARRAY	CONE
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	CONE
BS	R	C	FLOW CONDITIONS BEHIND COMPRESSION	CONE
BS6SQ	R	U	MACH NUMBER SQUARED	CONE
CASE	I	C	CASE NUMBER	CONE
CP	R	U	PRESSURE COEFFICIENT	CONE
CP	R	A	PRESSURE COEFFICIENT	CONE
DELTAR	R	U	SURFACE IMPACT ANGLE IN RADIANS	CONE
EMNS	R	U	MACH NUMBER NORMAL TO SHOCK	CONE
ERROR	I	C	ERROR FLAG	CONE
FS	R	C	FLOW CONDITIONS BEFORE COMPRESSION	CONE
G	R	U	RATIO OF SPECIFIC HEATS =1.4	CONE
IM	I	C	ELEMENT ROW NUMBER ARRAY	CONE
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	CONE
ISDET	I	A	DATA GENERATION CONTROL FLAG	CONE
K	I	C	NUMBER OF ELEMENTS	CONE
LS	I	C	NUMBER OF ELEMENTS	CONE
MACH	R	U	MACH NUMBER	CONE
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	CONE
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	CONE
NZ2	R	C	ELEMENT DIRECTION COSINF ARRAY-Z	CONE
PAGE	I	C	PAGE NUMBER	CONE
P2P1I	R	U	PRESSURE RATIO ACROSS COMPRESSION	CONE
R3	R	U	SHOCK ANGLE PARAMETER	CONE
TINT2	R	U	TEMPERATURE RATIO	CONE
TITLE	R	C	TITLE	CONE
X	R	U	SHOCK ANGLE PARAMETER	CONE
XCENT2	R	C	ELEMENT CENTROID ARRAY-X	CONE
YCENT2	R	C	ELEMENT CENTROID ARRAY-Y	CONE
ZCENT2	R	C	ELEMENT CENTROID ARRAY-Z	CONE

18. SUBROUTINE BLUNT (DECK AROQ)

This routine calculates the viscous forces on a blunt body including low density effects.

a. Algorithm

Checks for ideal or real gas option, then calculates local properties behind normal shock. Determines local viscous forces and calculates low density viscous-interaction effects.

b. Input/Output

IPRINT = 1, pertinent local and free-stream properties and viscous force coefficients will be printed. (This is intended for checkout only and IPRINT must be set within the program.)

c. Error

None

d. Subroutines Required

None

e. Argument List

(PFS, MACH, TFS, VIS, RHOFS, RB, RENO, TAU, IVISIN)

f. Length

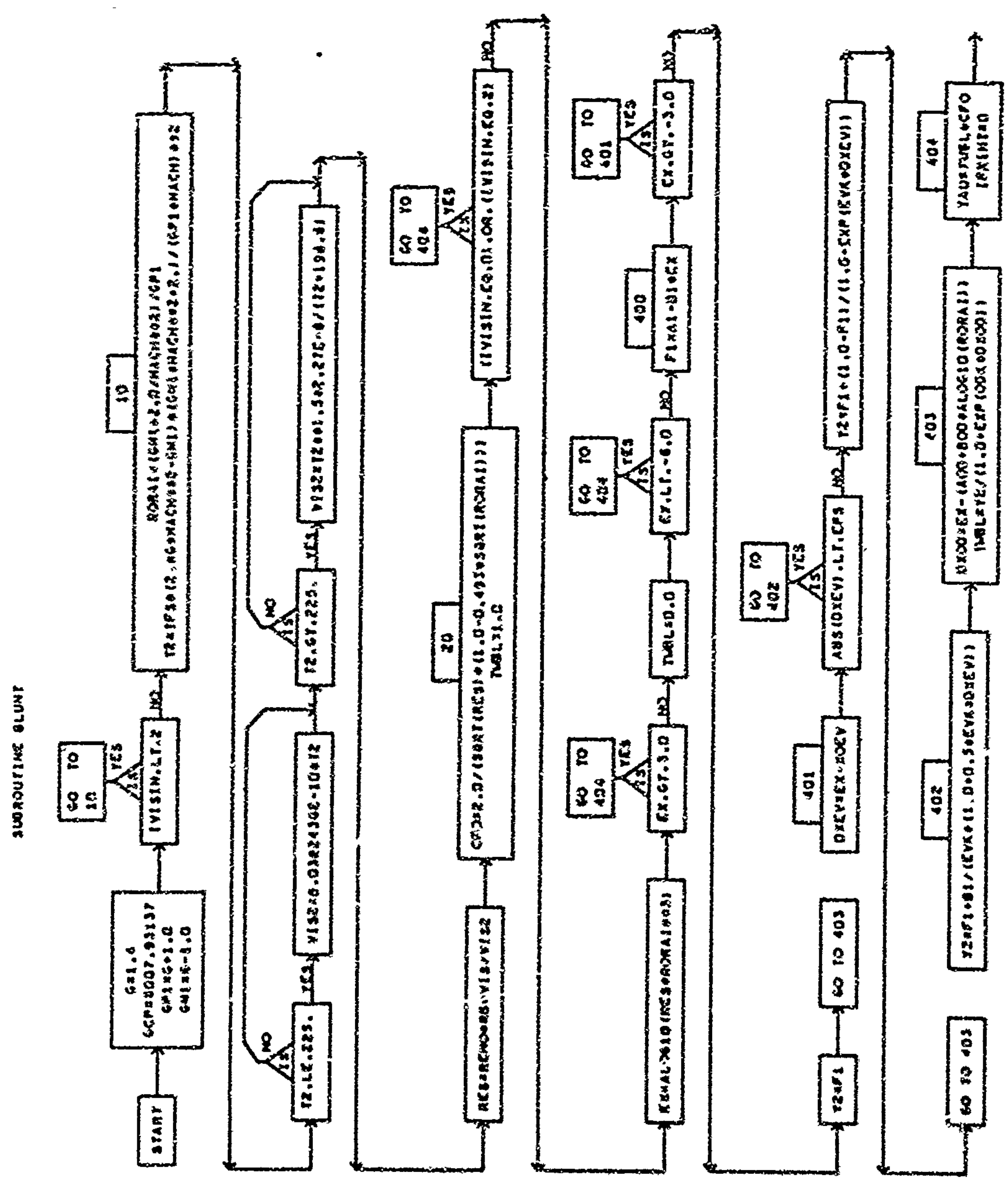
1596 bytes

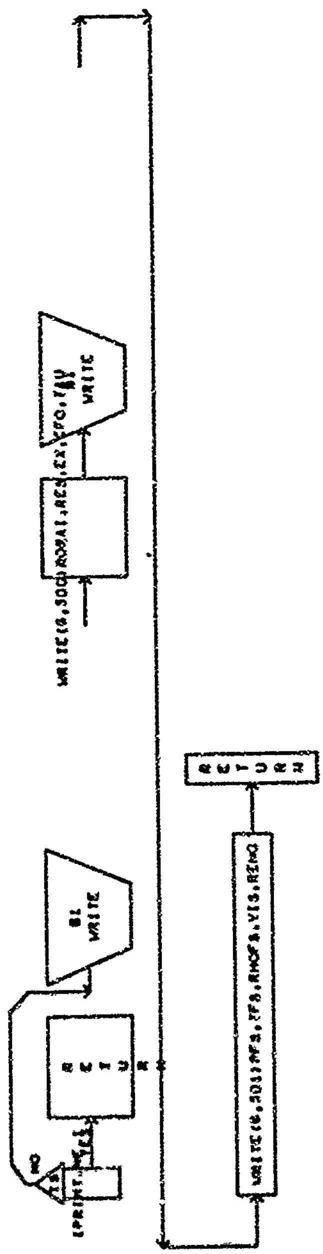
DECK AROQ

```
      SUBROUTINE BLUNT(PFS,MACH,TFS,VIS,RHOFS,RB,RENO,TAU,IVISIN)
C
C
C  THIS SUBROUTINE CALCULATES THE VISCOUS FORCES ON A BLUNT
C  FACED BODY FOLLOWING THE APPROACH SUGGESTED BY L. GOLDBERG IN
C  G. E. REPORT R66SD21 (SEE ALSO R65SD50).  THE STRAIGHT VISCOUS
C  FORCES ARE CALCULATED USING A SIMPLE CORRELATION FORMULA BASED ON
C  THE RESULTS OF SCALA AND GILBERT.  THE LOW DENSITY OR VISCOUS
C  INTERACTION EFFECTS ARE BASED ON NUMERICAL RESULTS OF HIGHER
C  ORDER BOUNDARY-LAYER SOLUTIONS.  THE SHEAR EFFECTS ARE A COMPLICATED
C  FUNCTION OF THE INVERSE DENSITY RATIO AND THE SHOCK REYNOLDS
C  NUMBER.  IN THE PRESENT CALCULATIONS THESE LOW DENSITY
C  EFFECTS ARE DETERMINED FROM A SET OF EXPONENTIAL FUNCTION
C  CURVES WHICH HAVE BEEN MATCHED TO THE NUMERICAL RESULTS.
C  CALCULATION OPTIONS ARE CONTROLLED BY THE FLAG IVISIN.
C
C        IVISIN     GAS      VISCOUS-INTERACTION
C           0       IDEAL          NO
C           1       IDEAL          YES
C           2       REAL           NO
C           3       REAL           YES
C
C  THE LAST TWO OPTIONS ARE NOT YET AVAILABLE.
C*******M*******D. N. SMYTH PROGRAM AUTHOR*****************************
C
C
C
C
C
      DATA  AOD,   BOD,     A1,     B1,     ODK,  XOEV,   EVK,    EPS  /
     1      1.0,  3.2907,  0.667,  1.1111, -2.0,  -0.3,  -1.80,   0.01 /
      REAL MACH
      G = 1.4
      GCP = 6007.93137
      GP1 = G + 1.0
```

```
DECK AROQ

      GM1 = G - 1.0
C
      IF (IVISIN.LT.2) GO TO 10
C
C  REAL GAS SOLUTION (TO BE ADDED).
C
C  IDEAL GAS SOLUTION.  ALL EQUATIONS FROM NACA TR-1135.
C  INVERSE DENSITY RATIO ACROSS NORMAL SHOCK.
   10 RORAI = (GM1 + 2.0/MACH**2)/GP1
C
C  TEMPERATURE BEHIND NORMAL SHOCK.
      T2 = TFS*(2.*G*MACH**2-GM1)*(GM1*MACH**2+2.)/(GP1*MACH)**2
C
C  CALCULATE VISCOSITY.
      IF (T2.LE.225.) VIS2 = 8.0382436E-10*T2
      IF (T2.GT.225.) VIS2 = T2**1.5*2.27E-8/(T2+198.6)
C
C  REYNOLDS NUMBER BEHIND NORMAL SHOCK.
      RES = RENO*RB*VIS/VIS2
C
C  CALCULATE SHEAR COEFFICIENT.
   20 CFO = 2.0/(SQRT(RES)*(1.0 - 0.495*SQRT(RORAI)))
C
C  CHECK IF LOW DENSITY VISCOUS-INTERACTION EFFECTS DESIRED.
      TWBL = 1.0
      IF((IVISIN.EQ.0).OR.(IVISIN.EC.2))GO TO 404
C
C  DETERMINE LOW DENSITY EFFECTS.  CALCULATE INDEPENDENT VARIABLE, EX.
      EX = ALOG10(RES*RORAI**3)
C
C  CHECK BOUNDARIES
      IF (EX.GT.3.0) GO TO 404
      TWBL = 0.0
      IF (EX.LT.-6.0) GO TO 404
C
C
```

```
DECK AROQ

  400 F1 = A1 - B1*EX
      IF  (EX.GT.-3.0) GO TO 401
      Y2 = F1
      GO TO 403
C
C
  401 DXEV = EX - XOEV
      IF  (ABS(DXEV).LT.EPS)GO TO 402
      Y2 = F1 + (1.0 - F1)/(1.0 - EXP(EVK*DXEV))
      GO TO 403
C
C
  402 Y2 = F1 + B1/(EVK*(1.0 + 0.5*EVK*DXEV))
  403 DXOD = EX - (AOD + BOD*ALOG10(RORAI))
      TWBL = Y2/(1.0 + EXP(ODK*DXOD))
C
C
  404 TAU = TWBL*CFO
C
C
C  THE FOLLOWING CARDS ARE FOR CHECKOUT ONLY (SET IPRINT = 1).
C
      IPRINT = 0
C
      IF (IPRINT.NE.1) RETURN
      WRITE(6,500) RORAI,RES,EX,CFO,TAU
  500 FORMAT(1H1, 2X7HRORAI =,E13.6,5X5HRES =,E13.6,5X11H(E**3)RES =,
     1      E13.6,5X5HCFO =,E13.6, 5X5HTAU =,E13.6)
      WRITE(6,501) PFS,TFS,RHOFS,VIS,RENO
  501 FORMAT(1H0,2X7H  PFS =,E13.6,5X5HTFS =,E13.6,5X11H    RHOFS =,
     1             E13.6,5X5HVIS =,E13.6, 4X6HRENO =,E13.6)
      RETURN
      END
```


SUBROUTINE BLUNT


WRITE
RETURN
WRITE
RETURN

```
SYMBOLS USED IN SUBROUTINE BLUNT

ADD      R  U  COEFFICIENT IN DEFINITION OF ODD ORIGIN                      BLUNT
A1       R  U  COEFFICIENT IN DEFINITION OF FIRST FUNCTION                  BLUNT
BOD      R  U  COEFFICIENT IN DEFINITION OF ODD ORIGIN                      BLUNT
B1       R  U  COEFFICIENT IN DEFINITION OF FIRST FUNCTION                  BLUNT
CFO      R  U  SKIN-FRICTION COEFFICIENT WITHOUT INTERACTION                BLUNT
DXEV     R  U  INCREMENT FROM ORIGIN, EVEN EXPONENTIAL                      BLUNT
EPS      R  U  TOLERANCE FOR EVEN EXPONENTIAL                               BLUNT
EVK      R  U  EVEN EXPONENTIAL CONSTANT                                    BLUNT
EX       R  U  INDEPENDENT VARIABLE                                         BLUNT
F1       R  U  FIRST FUNCTION OF EVEN EXPONENTIAL                           BLUNT
G        R  U  RATIO OF SPECIFIC HEATS                                      BLUNT
GCP      R  U  GAS SPECIFIC HEAT AT CONSTANT PRESSURE                       BLUNT
GM1      R  U  RATIO OF SPECIFIC HEATS MINUS ONE                            BLUNT
GP1      R  U  RATIO OF SPECIFIC HEATS PLUS ONE                             BLUNT
IVISIN   I  A  VISCOUS-INTERACTION CONTROL FLAG                             BLUNT
MACH     R  A  MACH NUMBER                                                  BLUNT
ODK      R  U  ODD EXPONENTIAL CONSTANT                                     BLUNT
PFS      R  A  FREE-STREAM PRESSURE                                         BLUNT
PXOD     R  U  INCREMENT FROM ORIGIN, ODD EXPONENTIAL                       BLUNT
RB       R  A  BODY NOSE RADIUS (FEET)                                      BLUNT
RENO     R  A  FREE-STREAM REYNOLDS NUMBER PER FOOT                         BLUNT
RES      R  U  SHOCK REYNOLDS NUMBER                                        BLUNT
RHOFS    R  A  FREE-STREAM DENSITY                                          BLUNT
RORAI    R  U  INVERSE DENSITY RATIO ACROSS NORMAL SHOCK                    BLUNT
TAU      R  A  SHEAR FORCE                                                  BLUNT
TFS      R  A  FREE-STREAM TEMPERATURE                                      BLUNT
TWBL     R  U  RATIO OF SHEAR FORCE WITH INTERACTION TO THAT WITHOUT        BLUNT
T2       R  U  TEMPERATURE BEHIND NORMAL SHOCK                              BLUNT
VIS      R  A  FREE-STREAM VISCOSITY                                        BLUNT
VIS2     R  U  VISCOSITY BEHIND NORMAL SHOCK                                BLUNT
XOEV     R  U  ORIGIN FOR EVEN EXPONENTIAL                                  BLUNT
Y2       R  U  SECOND FUNCTION OF ODD EXPONENTIAL                           BLUNT
```

19. SUBROUTINE TEMP (DECK AROR)

This routine uses an iterative procedure to calculate the surface equilibrium temperature for either an ideal gas or a real gas.

a. Algorithm

Calculate local and recovery enthalpy and general constants. Check for type of calculation desired (ideal or real gas, temperature input or calculate), proceed with iteration and determine local convective heating rate, reference Reynolds number and compressibility factors. Print local skin friction data and temperature iterations if required.

b. Input/Output

IPRINT = 1, temperature iterations and local skin friction data will be printed.

IPRINT = 2, only local skin friction data will be printed.

c. Error

None

d. Subroutines Required

QC

e. Argument List

(EL, TR, RE, TS, NW, MER, IPRINT, RT)

f. Length

3620 bytes

DECK AROR

```
      SUBROUTINE TEMP(EL,TR,RE,TS,NW,MER,IPRINT,RT)
C
C
C  CALCULATES WALL TEMPERATURE AND OTHER QUANTITIES NECESSARY
C  FOR SKIN FRICTION AND BOUNDARY LAYER CALCULATIONS.
C  HW IS CONTROL FLAG WITH OPTIONS GIVEN BY FOLLOWING MATRIX.
C
C        GAS       WALL TEMP. TYPE              METHOD FOR
C        TYPE    EQUIL   TAW   INPUT     LAMINAR    TURBULENT
C
C        IDEAL     0      1      2        REF. T       S-C
C
C        REAL      3      4      5        REF. H       S-C
C
C        IDEAL     6     ---     7        REF. T       REF. T
C
C        REAL      8     ---     9        REF. H       REF. H
C
C
C
C  BOTH LAMINAR (K = 1) AND TURBULENT (K = 2) DETERMINED.
C***************D. N. SMYTH PROGRAM AUTHOR*******************************
C
C
C
      DIMENSION TITLE(15),BS(8),FS(8),RE(2),TR(10),TS(2),RT(2),RF(2)
     1          ,ALP(20),BET(20),CCA(20),CCY(20),CCN(20),CCLL(20),
     2           CCLM(20),CCLN(20),CCL(20),CCD(20),CLOD(20),CF(20),
     3           CPS(20),ETACS(20),IS(10, 9),SURF(10, 8)
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),YCENT2( 300),
     1           ZCENT2( 300),AREA2( 300),IN( 300),IM( 300)
      COMMON CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1  AREA2,IN,IM,L,LS,FS,BS,ALP,BET,CCA,CCY,CCN,CCLL,CCLM,CCLN,CCL,
     2  CCD,CLOD,CF,CPS,ETACS,IS,SURF,NPRT
      COMMON /TEMPQC/HAW,H2,H1,HW,CKU,FC,FRX,RET,ELLOC,GCP,  TST1,ROMURA
```

```
DECK AROR
      COMMON /FLAG2/ITW,IHW,IFLOW,ITURB,CFTLOC                          AROR 0360
      REAL NX2,NY2,NZ2                                                  AROR 0370
      INTEGER CASE,PAGE,ERROR                                           AROR 0380
      DATA EPST,KTMAX,EMISS,PRAN,G                                      AROR 0390
     1     /5.E-4, 10, 0.8,  0.71 ,1.40                                 AROR 0400
     2     /                                                            AROR 0410
C                                                                       AROR 0420
C   SET UP GENERAL QUANTIES (GCP CONSISTENT WITH ATMOS).                AROR 0430
      GCP =  6007.93137                                                 AROR 0440
      H1 = GCP*FS(3)                                                    AROR 0450
      HTOT =         H1*(1. + 0.5*(G-1.)*FS(6)**2)                      AROR 0460
      H2 = HTOT - 0.5*BS(7)**2                                          AROR 0470
      ELLOC = EL                                                        AROR 0480
      RF(1) = SQRT(PRAN)                                                AROR 0490
      RF(2) = PRAN**(1./3.)                                             AROR 0500
      CKU = 0.332*FS(5)*SQRT(FS(8)/EL)/PRAN**(2./3.)*SQRT(BS(7)/FS(7))  AROR 0510
      RADK = 0.480E-12*EMISS*778.0                                      AROR 0520
C                                                                       AROR 0530
C   SET UP CONTROL FLAGS                                                AROR 0540
      ITURB = 1                                                         AROR 0550
      IF (NW.GT.5) ITURB = 2                                            AROR 0560
      NWI = NW                                                          AROR 0570
      ITW = 1                                                           AROR 0580
      IF (NWI.LT.3) GO TO 6                                             AROR 0590
      ITW = 2                                                           AROR 0600
      NWI = NWI - 3                                                     AROR 0610
      IF (NWI.LT.3) GO TO 6                                             AROR 0620
      IF (NWI.LT.5) GO TO 4                                             AROR 0630
      NWI = NWI - 5                                                     AROR 0640
      GO TO 5                                                           AROR 0650
    4 ITW = 1                                                           AROR 0660
      NWI = NWI - 3                                                     AROR 0670
    5 IF (NWI.NE.0) NWI = 2                                             AROR 0680
C                                                                       AROR 0690
C   MAJOR LOOP FOR CALCULATING LAMINAR (K=1) AND TURBULENT (K=2) FLOW.  AROR 0700
    6 DO 1000 K = 1,2                                                   AROR 0710
```

```
DECK AROR

      IF (IPRINT.NE.1) GO TO 302                                        AROR 0720
      WRITE(6,301)                                                      AROR 0730
      NPRT = NPRT + 1                                                   AROR 0740
  301 FORMAT(1H )                                                       AROR 0750
  302 KT = 0                                                            AROR 0760
      IFLOW = K                                                         AROR 0770
      IHW = 0                                                           AROR 0780
      HAW = H2 + RF(K)*(HTOT - H2)                                      AROR 0790
      TC1 = 100.                                                        AROR 0800
      IF (NWI - 1) 9,7,8                                                AROR 0810
    7 HW = HAW                                                          AROR 0820
      IHW = 1                                                           AROR 0830
      GO TO 21                                                          AROR 0840
    8 TC1 = TR(K+4)                                                     AROR 0850
      IF (TC1.GT.7000.) TC1 = 7000.                                     AROR 0860
      GO TO 21                                                          AROR 0870
    9 ITW = 1                                                           AROR 0880
C  NOTE, FOR REAL GAS EQUILIBRIUM TW IDEAL GAS DONE FIRST.              AROR 0890
      TR1 = TC1                                                         AROR 0900
      QC1 = QC(TC1)                                                     AROR 0910
      QR1 = RADK*1.E+8                                                  AROR 0920
      QR2 = QC1                                                         AROR 0930
      TR2 = (QR2/RADK)**0.25                                            AROR 0940
      TC2 = HAW/GCP                                                     AROR 0950
      IF (TR2 .LT.TC2) GO TO 3                                          AROR 0960
      QC2 = 0.0                                                         AROR 0970
      TR2 = TC2                                                         AROR 0980
      QR2 = RADK*TR2**4                                                 AROR 0990
      GO TO 10                                                          AROR 1000
    3 TC2 = TR2                                                         AROR 1010
      QC2 = QC(TC2)                                                     AROR 1020
C                                                                       AROR 1030
C  ITERATION CYCLE FOR TW                                               AROR 1040
   10 KT = KT + 1                                                       AROR 1050
C**CHECK IF TEMPERATURE ITERATIONS TO BE PRINTED.                       AROR 1060
      IF (IPRINT.EQ.1)                                                  AROR 1070
```

DECK AROR

```
      1WRITE(6,300) KT,TC1,TR1,TC2,TR2,ITW,QC1,QR1,QC2,QR2
  300 FORMAT(1H ,2X4HKT =,I4,5X5HTC1 =,E13.6,5X5HTR1 =,E13.6,
     1       5X5HTC2 =,E13.6,5X5HTR2 =,E13.6, /3X4HITW=,I4,
     2       5X5HQC1 =,E13.6,5X5HQR1 =,E13.6,5X5HQC2 =,E13.6,
     3       5X5HQR2 =,E13.6)
C
C  CHECK IF ALLOWABLE NUMBER OF ITERATIONS EXCEEDED.
      IF (KT.GT.KTMAX) GO TO 22
      DQC = QC1-QC2
      DQR = QR2-QR1
      DTC = TC1-TC2
      DTR = TR2-TR1
C  LINEAR SOLUTION (OR INTERCEPT FOR NEXT GUESS.
      TC1 = ((QR1*TR2 - QR2*TR1)*DTC + (QC1*TC2 - QC2*TC1)*DTR)/
     1      (DQC*DTR - DQR*DTC)
C
C  CALCULATE HEATING RATES AND CHECK CONVERGENCE.
      IF (TC1.LT.0.0) GO TO 81
   11 TR1 = TC1
      QR1 = RADK*TR1**4
      QC1 = QC(TC1)
      IF (ABS(1. - QC1/QR1).LE.EPST) GO TO 12
C  NO SOLUTION, INITIATE NEXT CYCLE.
      IF (QC1.GT.0.0) GO TO 83
C
C  QC1 NEQATIVE, SPECIAL INITIALIZATION USED.
   81 KSUB = 1
   82 QC2 = 0.0
      TC2 = HAW/GCP
      TR2 = TC2
      QR2 = RADK*TR2**4
      IF (KSUB.NE.1) GO TO 10
      TC1 = TC2
   80 TC1 = 0.5*TC1
      QC1 = QC(TC1)
      TR1 = TC1
```

TEMP

```
DECK AROR

      QR1 = RADK*TR1**4                                                 AROR  1440
      IF (QC1.GT.QR1) GO TO 10                                          AROR  1450
      IF (KSUB.EQ.5) GO TO 10                                           AROR  1460
      KSUB = KSUB + 1                                                   AROR  1470
      GO TO 80                                                          AROR  1480
C                                                                       AROR  1490
C  QC1 POSITIVE, CONTINUE INITIALIZATION OF NEXT CYCLE.                 AROR  1500
   83 QR2 = QC1                                                         AROR  1510
      TR2 = (QR2/RADK)**0.25                                            AROR  1520
      TC2 = TR2                                                         AROR  1530
      QC2 = QC(TC2)                                                     AROR  1540
      IF (ABS(1. - QC2/QR2).LE.EPST) GO TO 84                           AROR  1550
      IF (QC2.GT.0.0) GO TO 10                                          AROR  1560
      KSUB = 2                                                          AROR  1570
      GO TO 82                                                          AROR  1580
C                                                                       AROR  1590
C  SOLUTION OBTAINED                                                    AROR  1600
   84 TC1 = TC2                                                         AROR  1610
C                                                                       AROR  1620
C  CHECK IF REAL GAS SOLUTION DESIRED.                                  AROR  1630
   12 IF ((ITW.EQ.2).OR.(NW.LT.3)) GC TO 21                             AROR  1640
      IF ((NW.EQ.6).OR.(NW.EQ.7)) GO TO 21                              AROR  1650
C                                                                       AROR  1660
C  DETERMINE REAL GAS SOLUTION                                          AROR  1670
      ITW = 2                                                           AROR  1680
      KT = 0                                                            AROR  1690
      GO TO 11                                                          AROR  1700
C                                                                       AROR  1710
C  EXCEEDED ALLOWABLE ITERATIONS (KTMAX), AVERAGE LAST TWO VALUES       AROR  1720
   22 CONTINUE                                                          AROR  1730
      TC1 = (TC1 + TC2)*0.5                                             AROR  1740
      GO TO 12                                                          AROR  1750
C                                                                       AROR  1760
C  CALCULATE QC AT FINAL TW VALUE TO SET QUANTITIES IN COMMON.          AROR  1770
   21 QC1 = QC(TC1)                                                     AROR  1780
   30 TR(K+4) = TC1                                                     AROR  1790
```

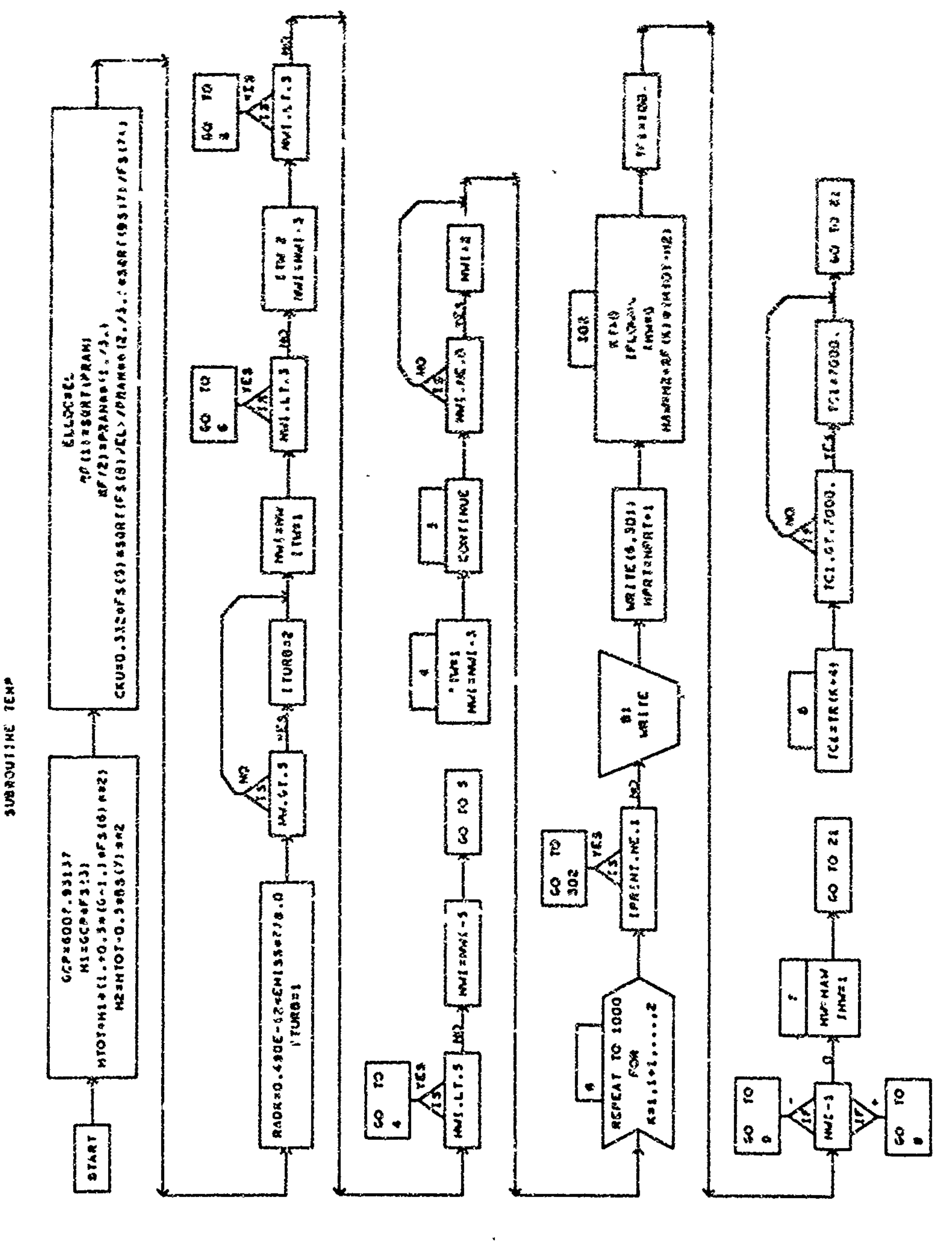
DECK AROR

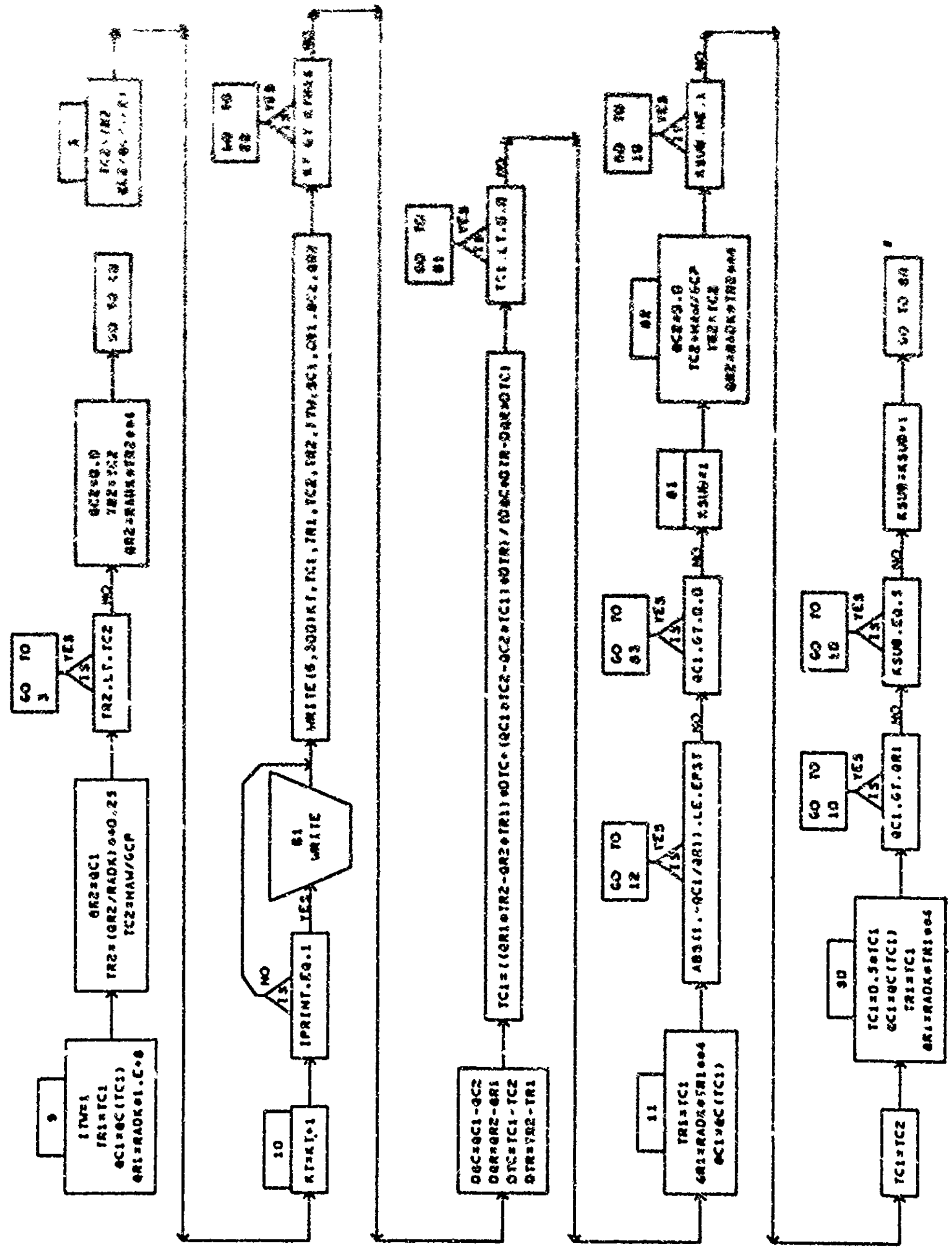
```
      RT(K) = HAW/GCP                                                   AROR 1800
      TR(K+6) = HW                                                      AROR 1810
      TR(K+8) = HAW                                                     AROR 1820
      TS(K) = TST1*FS(3)                                                AROR 1830
C  RT AND TS ARE CORRECT ONLY FOR AN IDEAL GAS.                         AROR 1840
      GO TO (23,24),K                                                   AROR 1850
   23 CFLLOC = CKU*ROMURA*2.0*PRAN**(2./3.)/(BS(1)*BS(7))               AROR 1860
      RE(K) = (0.664*BS(3)/(TS(K)*CFLLOC))**2                           AROR 1870
      NW1 = 0                                                           AROR 1880
C                                                                       AROR 1890
C  CHECK IF PRINTOUT OF LOCAL SKIN FRICTION CHARACTERISTICS DESIRED.    AROR 1900
      IF  ((IPRINT.LT.1).OR.(IPRINT.GT.2)) GO TO 1000                   AROR 1910
C                                                                       AROR 1920
C  THE FOLLOWING CARDS ARE FOR LOCAL CF PRINTOUT ONLY.                  AROR 1930
C  LAMINAR FLOW                                                         AROR 1940
      CF1RE1 = 0.664*SQRT((BS(7)/FS(7))**3)*ROMURA                      AROR 1950
      CF1 = CF1RE1/SQRT(FS(8)*ELLOC)                                    AROR 1960
      RORA = ROMURA**2                                                  AROR 1970
      HAWH1 = HAW/H1                                                    AROR 1980
      NW1 = NW + 1                                                      AROR 1990
   39 NPRT = NPRT + 2                                                   AROR 2000
      GO TO (40,40,40,41,41,41,42,42,43,43),NW1                         AROR 2010
   40 WRITE(6,50) NW                                                    AROR 2020
   50 FORMAT(1H0,2X3HNW=,I2,3X32HIDEAL GAS, REF. T/S-C  SOLUTION.)      AROR 2030
      GO TO 44                                                          AROR 2040
   41 WRITE(6,51) NW                                                    AROR 2050
   51 FORMAT(1H0,2X3HNW=,I2,3X32HREAL  GAS, REF. H/S-C  SOLUTION.)      AROR 2060
      GO TO 44                                                          AROR 2070
   42 WRITE(6,52) NW                                                    AROR 2080
   52 FORMAT(1H0,2X3HNW=,I2,3X35HIDEAL GAS, REF. T/REF. T  SOLUTION.)   AROR 2090
      GO TO 44                                                          AROR 2100
   43 WRITE(6,53) NW                                                    AROR 2110
   53 FORMAT(1H0,2X3HNW=,I2,3X35HREAL  GAS, REF. H/REF. H  SOLUTION.)   AROR 2120
   44 WRITE(6,217) KT,TC1,CF1,CF1RE1,RORA,TST1,HAWH1                    AROR 2130
      NPRT = NPRT + 1                                                   AROR 2140
  217 FORMAT(1H , 2X3HKT=, I2, 3X6HTWEQ =,F7.1,1HR, 3X5HCF1 =,E13.6,    AROR 2150
```

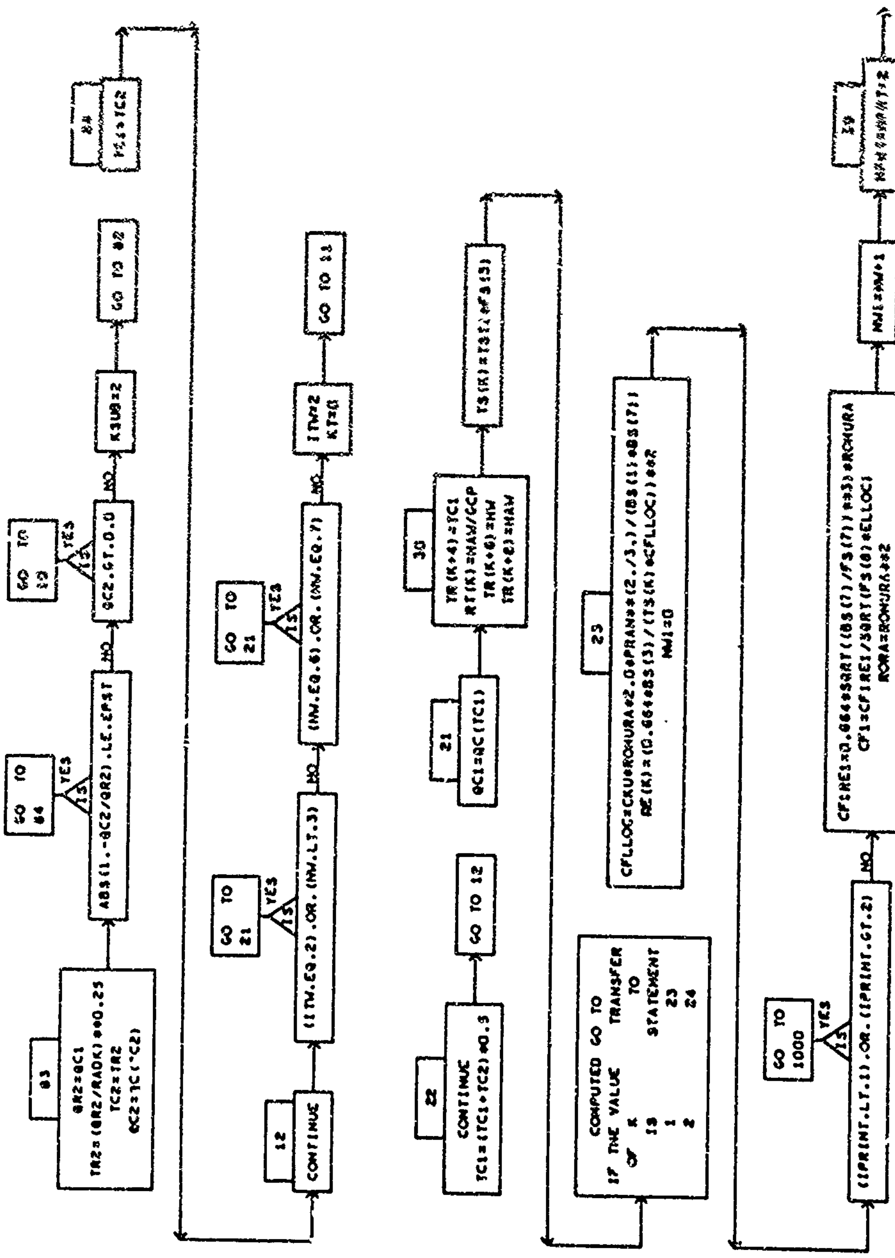

TEMP

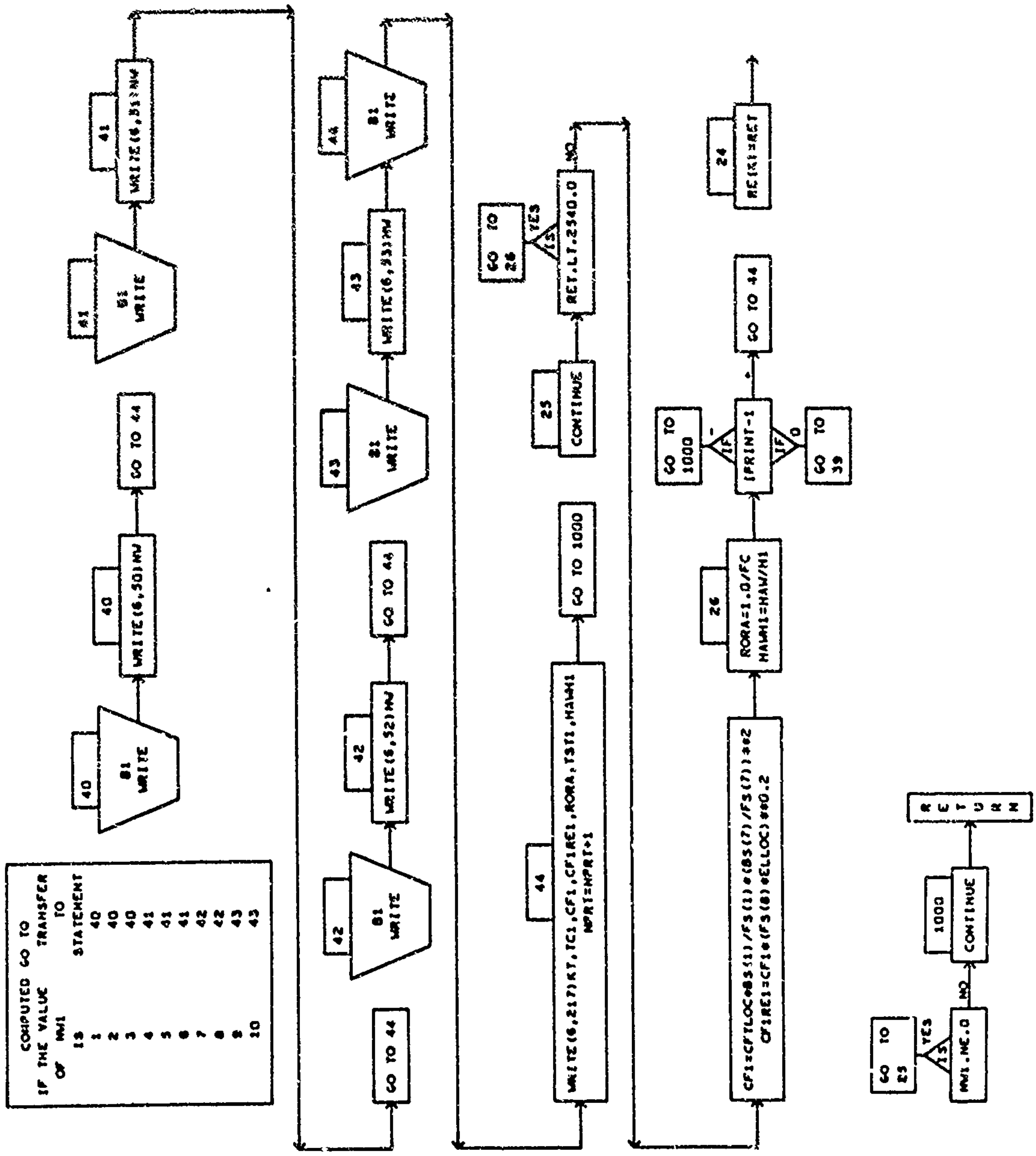
DECK AROR

```
      1        3X10HCF1(RE1) =,F9.6,3X9H ROMURA =,F9.5, 3X7HH*/H1 =,F9.4,
      2         3X8HHAW/H1 =,F9.4)
        GO TO 1000
C
C   TURBULENT FLOW
    25 IF (RET.LT.2540.C) GO TO 26
        CF1 = CFTLOC*BS(1)/FS(1)*(BS(7)/FS(7))**2
        CF1RE1 = CF1*(FS(8)*ELLOC)**0.2
    26 RORA = 1.0/FC
        HAWH1 = HAW/H1
        IF (IPRINT - 1) 1000,39,44
C
C   THIS ENDS PRINTOUT CARDS.
    24 RE(K) = RET
        IF (NW1.NE.0) GO TO 25
  1CCO CONTINUE
        RETURN
        END
```


SUBROUTINE TEMP
START
CONTINUE
WRITE
GO TO 302




83
QR2=QC1
TR2=(QR2/RADK)**0.25
TC2=TR2
QC2=QC(TC2)
ABS(1.-QC2/QR2).LE.EPST
IS
YES
GO TO 84
NO
QC2.GT.0.0
IS
YES
GO TO 19
NO
KSUB=2
GO TO 82
84
12
CONTINUE
((TW.EQ.2).OR.(NW.LT.3)
IS
YES
GO TO 21
NO
(NW.EQ.6).OR.(NW.EQ.7)
IS
YES
GO TO 21
NO
ITW=2
KT=0
GO TO 11
22
CONTINUE
TC1=(TC1+TC2)*0.5
GO TO 12
21
QC1=QC(TC1)
30
TR(K+4)=TC1
RT(K)=HAW/CCP
TR(K+6)=HW
TR(K+8)=HAW
COMPUTED GO TO
IF THE VALUE
OF K
IS
1
2
TRANSFER
TO
STATEMENT
23
24
23
CFLLOC=CKU*ROHURA*2.0*PRAN**(2./3.)/(BS(1)*BS(7))
RE(K)=(0.664*BS(3)/(TS(K)*CFLLOC))**2
NW1=0
(IPRINT.LT.1).OR.(IPRINT.GT.2)
IS
YES
GO TO 1000
NO
CF1RE1=0.664*SQRT((BS(7)/FS(7))**3)*ROHURA
CF1=CF1RE1/SQRT(FS(8)*ELLOC)
RORA=ROHURA**2
HAWH1=HAW/H1
NW1=NW1+1
19


COMPUTED GO TO
IF THE VALUE OF NW1 IS
TRANSFER TO STATEMENT
1 40
2 40
3 40
4 41
5 41
6 41
7 42
8 42
9 43
10 43
40
B1 WRITE
40
WRITE(6,50)NW
GO TO 44
41
B1 WRITE
41
WRITE(6,51)NW
42
B1 WRITE
42
WRITE(6,52)NW
GO TO 44
GO TO 44
43
B1 WRITE
43
WRITE(6,53)NW
44
B1 WRITE
44
WRITE(6,217)KT,TC1,CF1,CF1RE1,RORA,TST1,MAWM1
NPRI=NPRI+1
GO TO 1000
25
CONTINUE
RET.LT.2540.0
IS
YES
GO TO 26
NO
CF1=CF1LOC*BS(1)/FS(1)*(BS(7)/FS(7))**2
CF1RE1=CF1*(FS(8)*ELLOC)**0.2
26
RORA=1.0/FC
MAWM1=MAW/M1
IFRINT-1
IF
GO TO 1000
IF
GO TO 39
GO TO 44
24
RET(K)=RET
NW1.NE.0
IS
YES
GO TO 25
NO
1000
CONTINUE
RETURN

SYMBOLS USED IN SUBROUTINE TEMP

ALP	R	C	ANGLE OF ATTACK ARRAY	TEMP
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	TEMP
BET	R	C	YAW ANGLE ARRAY	TEMP
BS	R	C	FLOW CONDITIONS BEHIND THE SHOCK OR EXPANSION	TEMP
CASE	I	C	CASE NUMBER	TEMP
CCA	R	C	AXIAL FORCE COEFFICIENT ARRAY	TEMP
CCD	R	C	DRAG COEFFICIENT ARRAY	TEMP
CCL	R	C	LIFT COEFFICIENT ARRAY	TEMP
CCLL	R	C	ROLLING MOMENT COEFFICIENT ARRAY	TEMP
CCLM	R	C	PITCHING MOMENT COEFFICIENT ARRAY	TEMP
CCLN	R	C	YAWING MOMENT COEFFICIENT ARRAY	TEMP
CCN	R	C	NORMAL FORCE COEFFICIENT ARRAY	TEMP
CCY	R	C	SIDE FORCE COEFFICIENT ARRAY	TEMP
CF	R	C	SKIN FRICTION CONTRIBUTION TO AXIAL FORCE ARRAY	TEMP
CFLLOC	R	U	LOCAL LAMINAR SKIN-FRICTION COEFFICIENT	TEMP
CFTLOC	R	C	LOCAL TURBULENT SKIN-FRICTION COEFFICIENT	TEMP
CFI	R	U	SKIN FRICTION COEFFICIENT REFERENCED TO FREE-STREAM	TEMP
CFIREI	R	U	SKIN FRICTION PARAMETER	TEMP
CKU	R	C	LAMINAR FLOW FLIGHT CONDITION CONSTANT	TEMP
CLOD	R	C	LIFT TO DRAG RATIO ARRAY	TEMP
CPS	R	C	ARRAY FOR NEWTONIAN CORRECTION FACTOR, K	TEMP
DQC	R	U	DIFFERENCE IN CONVECTIVE HEATING RATES	TEMP
DQR	R	U	DIFFERENCE IN RADIATION HEATING RATES	TEMP
DTC	R	U	DIFFERENCE IN CONVECTIVE TEMPERATURES	TEMP
DTR	R	U	DIFFERENCE IN RADIATION TEMPERATURES	TEMP
EL	R	A	REFERENCE LENGTH	TEMP
ELLOC	R	C	REFERENCE LENGTH (=EL)	TEMP
EMISS	R	U	EMISSIVITY	TEMP
EPST	R	U	TOLERANCE OF TEMPERATURE ITERATIONS	TEMP
ERROR	I	C	ERROR FLAG	TEMP
ETACS	R	C	SAVED VALUES OF PRANDTL-MEYER CORRECTION FACTOR	TEMP
FC	R	C	TURBULENT FLOW, SKIN-FRICTION COMPRESSIBILITY FACTOR	TEMP
FRX	R	C	TURBULENT FLOW, REYNOLDS NUMBER COMPRESSIBILITY FACTOR	TEMP
FS	R	C	FLOW CONDITIONS BEFORE THE SHOCK OR EXPANSION	TEMP
G	R	U	RATIO OF SPECIFIC HEATS	TEMP
GCP	R	C	GAS SPECIFIC HEAT AT CONSTANT PRESSURE	TEMP

TEMP

SYMBOLS USED IN SUBROUTINE TEMP

HAW	R	C	ADIABATIC-WALL ENTHALPY	TEMP
HAWH1	R	U	ADIABATIC-WALL TO FREE-STREAM ENTHALPY RATIO	TEMP
HTOT	R	U	TOTAL ENTHALPY	TEMP
HW	R	C	WALL ENTHALPY	TEMP
H1	R	C	FREE-STREAM ENTHALPY	TEMP
H2	R	C	LOCAL ENTHALPY	TEMP
IFLOW	I	C	LAMINAR (=1) OR TURBULENT (=2) FLOW FLAG	TEMP
IHW	I	C	WALL ENTHALPY FLAG	TEMP
IM	I	C	ELEMENT ROW NUMBER ARRAY	TEMP
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	TEMP
IPRINT	I	A	PRINT FLAG	TEMP
IS	I	C	SKIN FRICTION FLAG DATA ARRAY	TEMP
ITURB	I	C	TURBULENT FLOW FLAG	TEMP
ITW	I	C	IDEAL GAS (=1) OR REAL GAS(=2) FLAG	TEMP
K	I	U	FLAG (=1 LAMINAR, =2 TURBULENT)	TEMP
KSUB	I	U	SECONDARY COUNTER IN TEMPERATURE ITERATIONS	TEMP
KT	I	U	TEMPERATURE ITERATION COUNTER	TEMP
KTMAX	I	U	MAXIMUM NUMBER OF TEMPERATURE ITERATIONS	TEMP
L	I	C	NUMBER OF ELEMENTS	TEMP
LS	I	C	NUMBER OF ELEMENTS	TEMP
MER	I	A	ERROR FLAG	TEMP
NPRT	I	C	PRINT COUNTER	TEMP
NW	I	A	TEMPERATURE CALCULATION CONTROL FLAG	TEMP
NWI	I	U	TEMPERATURE CALCULATION CONTROL FLAG	TEMP
NW1	I	U	TEMPERATURE CALCULATION PRINT CONTROL FLAG	TEMP
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	TEMP
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	TEMP
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	TEMP
PAGE	I	C	PAGE NUMBER	TEMP
PRAN	R	U	PRANDTL NUMBER	TEMP
QC1	R	U	CONVECTIVE HEATING RATE AT TC1	TEMP
QC2	R	U	CONVECTIVE HEATING RATE AT TC2	TEMP
QR1	R	U	RADIATION HEATING RATE AT TR1	TEMP
QR2	R	U	RADIATION HEATING RATE AT TR1	TEMP
RADK	R	U	RADIATION CONSTANT	TEMP
RE	T	A	REFERENCE REYNOLDS NUMBER	TEMP

SYMBOLS USED IN SUBROUTINE TEMP

```
RET      R  C  TURBULENT FLOW REYNOLDS NUMBER AT REFERENCE CONDITION          TEMP
RF       R  D  RECOVERY FACTOR                                                TEMP
RCMURA   R  C  SQUARE-ROOT OF REFERENCE DENSITY-VISCOSITY RATIO               TEMP
RORA     R  J  REFERENCE TO FREE-STREAM DENSITY-VISCOSITY RATIO               TEMP
RT       R  A  RECOVERY TEMPERATURE                                           TEMP
SURF     R  C  SKIN FRICTION DATA ARRAY                                       TEMP
TC1      R  U  FIRST VALUE OF CONVECTIVE TEMPERATURE                          TEMP
TC2      R  U  SECOND VALUE OF CONVECTIVE TEMPERATURE                         TEMP
TITLE    R  C  TITLE                                                          TEMP
TR       R  A  FLIGHT CONDITION AND SKIN FRICTION DATA ARRAY                  TEMP
TR1      R  U  FIRST VALUE OF RADIATION TEMPERATURE                           TEMP
TR2      R  U  SECOND VALUE OF RADIATION TEMPERATURE                          TEMP
TS       R  A  REFERENCE TEMPERATURE (T STAR)                                 TEMP
TST1     R  C  REFERENCE TO FREE-STREAM TEMPERATURE (OR ENTHALPY) RATIO       TEMP
XCENT2   R  C  QUADRILATERAL CENTROID ARRAY-X                                 TEMP
YCENT2   R  C  QUADRILATERAL CENTROID ARRAY-Y                                 TEMP
ZCENT2   R  C  QUADRILATERAL CENTROID ARRAY-Z                                 TEMP
```

SYMBOLS USED IN SUBROUTINE TEMP

RET	R	C	TURBULENT FLOW REYNOLDS NUMBER AT REFERENCE CONDITION	TEMP
RF	R	D	RECOVERY FACTOR	TEMP
RCMURA	R	C	SQUARE-ROOT OF REFERENCE DENSITY-VISCOSITY RATIO	TEMP
RORA	R	J	REFERENCE TO FREE-STREAM DENSITY-VISCOSITY RATIO	TEMP
RT	R	A	RECOVERY TEMPERATURE	TEMP
SURF	R	C	SKIN FRICTION DATA ARRAY	TEMP
TC1	R	U	FIRST VALUE OF CONVECTIVE TEMPERATURE	TEMP
TC2	R	U	SECOND VALUE OF CONVECTIVE TEMPERATURE	TEMP
TITLE	R	C	TITLE	TEMP
TR	R	A	FLIGHT CONDITION AND SKIN FRICTION DATA ARRAY	TEMP
TR1	R	U	FIRST VALUE OF RADIATION TEMPERATURE	TEMP
TR2	R	U	SECOND VALUE OF RADIATION TEMPERATURE	TEMP
TS	R	A	REFERENCE TEMPERATURE (T STAR)	TEMP
TST1	R	C	REFERENCE TO FREE-STREAM TEMPERATURE (OR ENTHALPY) RATIO	TEMP
XCENT2	R	C	QUADRILATERAL CENTROID ARRAY-X	TEMP
YCENT2	R	C	QUADRILATERAL CENTROID ARRAY-Y	TEMP
ZCENT2	R	C	QUADRILATERAL CENTROID ARRAY-Z	TEMP

20. FUNCTION QC (DECK AROS)

This routine calculates the aerodynamic heating at the given wall temperature.

a. Algorithm

Tests for laminar or turbulent flow, for reference method or Spalding-Chi, and for ideal or real gas. Calculates convective heating rate and sets certain quantities in common.

b. Input/Output

None

c. Error

None

d. Subroutines Required

ROMU (three entries, ROMU, ROW, ENTHAL)

e. Argument List

(TW)

f. Length

2132 bytes

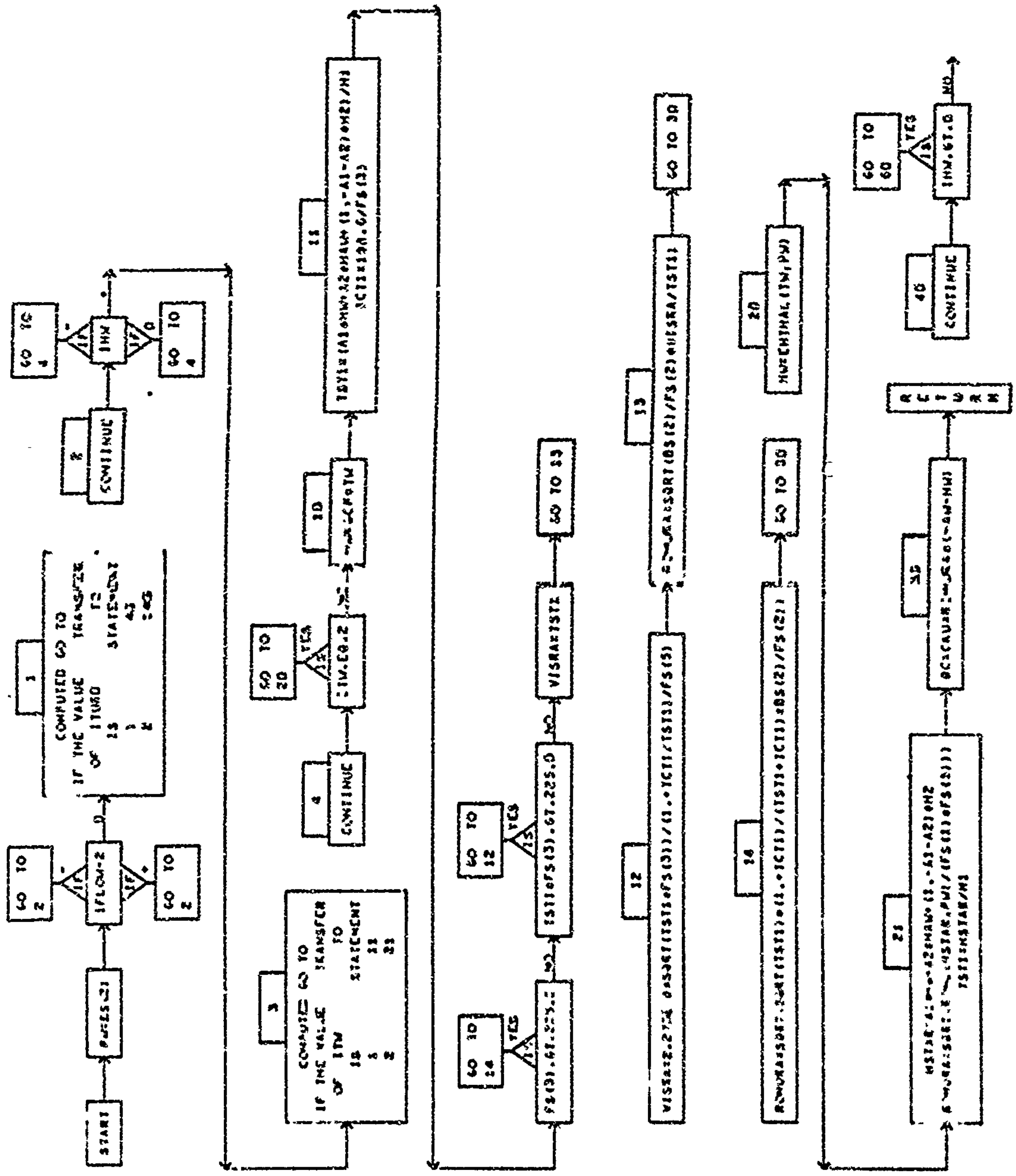
DECK AROS

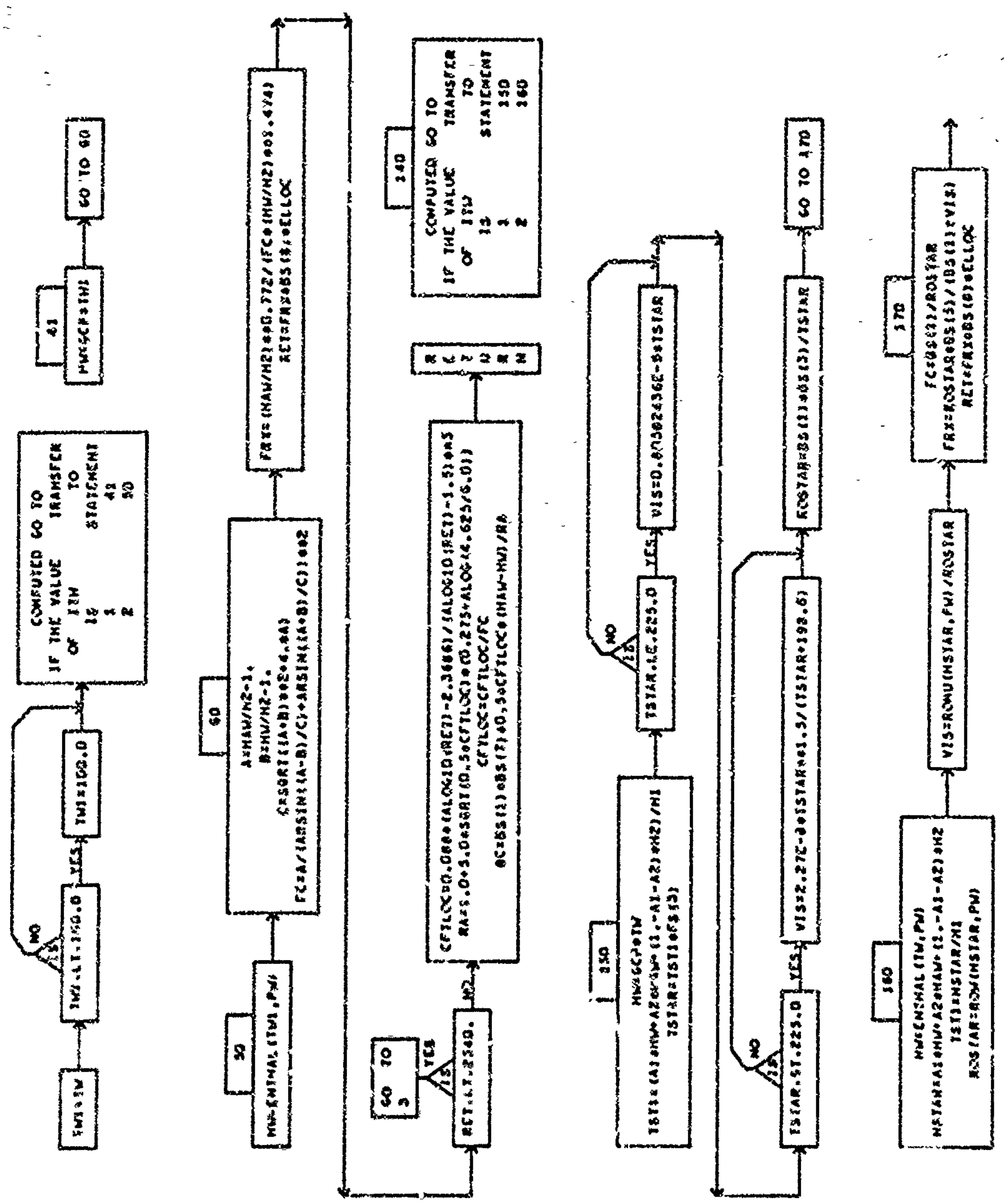
```
      FUNCTION  QC(TW)
C
C
C  CALCULATES THE AERODYNAMIC HEATING AT THE GIVEN
C  WALL TEMP.(TW) IN LAMINAR (IFLOW = 1) AND TURBULENT
C  (IFLOW = 2) FLOW OF EITHER AN IDEAL GAS (ITW = 1)
C  OR A REAL GAS (ITW = 2).  REFERENCE TEMPERATURE OR
C REFERENCE ENTHALPY USED FOR LAMINAR FLOW.  SPALDING-CHI (ITURB = 1)
C  OR REFERENCE TEMPERATURE/REFERENCE ENTHALPY (ITURB = 2) USED
C  FOR TURBULENT FLOW.
C***************D. N. SMYTH PROGRAM AUTHOR*******************************
C
C
C
      DIMENSION TITLE(15),BS(8),FS(8)
      DIMENSION NX2( 300),NY2( 300),NZ2( 300),XCENT2( 300),YCENT2( 300),
     1          ZCENT2( 300),AREA2( 300),IN( 300),IM(300)
      COMMON CASE,TITLE,PAGE,ERROR,NX2,NY2,NZ2,XCENT2,YCENT2,ZCENT2,
     1       AREA2,IN,IM,L,LS,FS,BS
      COMMON /TEMPQC/HAW,H2,H1,HW,CKU,FC,FRX,RET,ELLOC,GCP,  TST1,ROMURA
      COMMON /FLAG2/ITW,IHW,IFLOW,ITURB,CFTLOC
      REAL NX2,NY2,NZ2
      INTEGER CASE,PAGE,ERROR
C
C  MONAGHAN REFERENCE CONDITION COEFFICIENTS (PR = 0.71)
      DATA A1,A2/0.5825,0.1875/
      PW = BS(2)
C
C  CHECK FOR LAMINAR OR TURBULENT.
      IF (IFLOW.EQ.2) GO TO (40,140), ITURB
C
C
C  LAMINAR FLOW.  CHECK IF ENTHALPY INPUT.
      IF (IHW.GT.0) GO TO (11,21), ITW
      GO TO (10,20), ITW
```

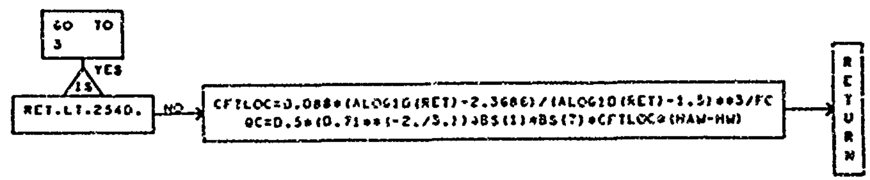
```
DECK AROS

C
C  REFERENCE TEMP. SOLUTION.
   10 HW = GCP*TW
   11 TST1 = (A1*HW + A2*HAW + (1.-A1-A2)*H2)/H1
      TCT1 = 198.6/FS(3)
      IF (FS(3).GT.225.0) GO TO 14
      IF (TST1*FS(3).GT.225.0) GO TO 12
      VISRA = TST1
      GO TO 13
   12 VISRA = 2.270E-8*SQRT(TST1*FS(3))/(1.+ TCT1/TST1)/ FS(5)
   13 ROMURA = SQRT(BS(2)/FS(2)*VISRA/TST1)
      GO TO 30
   14 ROMURA = SQRT(SQRT(TST1)*(1.+TCT1)/(TST1+TCT1)*BS(2)/FS(2))
      GO TO 30
C
C  REFERENCE ENTHALPY SOLUTION.
   20 HW = ENTHAL(TW,PW)
   21 HSTAR = A1*HW + A2*HAW + (1.-A1-A2)*H2
      ROMURA = SQRT(ROMU(HSTAR,PW)/(FS(1)*FS(5)))
      TST1 = HSTAR/H1
   30 QC = CKU*ROMURA*(HAW - HW)
      RETURN
C
C
C  TURBULENT FLOW,  SPALDING-CHI METHOD
   40 IF (IHW.GT.0) GO TO 60
      TW1 = TW
      IF(TW1.LT.100.0) TW1= 100.0
      GO TO (41,50),ITW
C  IDEAL GAS SOLUTION
   41 HW = GCP*TW1
      GO TO 60
C  REAL GAS SOLUTION
   50 HW = ENTHAL(TW1,PW)
   60 A = HAW/H2 - 1.
      B = HW/H2 - 1.
```

DECK AROS

```
      C = SQRT((A+B)**2 + 4.*A)                                        AROS  0720
      FC = A/(ARSIN((A-B)/C) + ARSIN((A+B)/C))**2                      AROS  0730
      FRX = (HAW/H2)**0.772/(FC*(HW/H2)**1.474)                        AROS  0740
      RET = FRX*BS(8)*ELLOC                                            AROS  0750
      IF (RET.LT.2540.) GO TO (11,21), ITW                             AROS  0760
      CFTLOC = 0.088*(ALOG10(RET) - 2.3686)/(ALOG10(RET)-1.5)**3       AROS  0770
      RA = 1.0 + 5.0*SQRT(0.5*CFTLOC)*(0.275 + ALOG(4.625/6.0))        AROS  0780
      CFTLOC = CFTLOC/FC                                               AROS  0790
      QC = BS(1)*BS(7)*0.5*CFTLOC*(HAW - HW)/RA                        AROS  0800
      RETURN                                                           AROS  0810
C                                                                      AROS  0820
C  TURBULENT FLOW,  REFERENCE METHOD                                   AROS  0830
  140 GO TO (150,160), ITW                                             AROS  0840
C  IDEAL GAS - REFERENCE TEMPERATURE                                   AROS  0850
  150 HW = GCP*TW                                                      AROS  0860
      TST1 = (A1*HW + A2*HAW + (1.-A1-A2)*H2)/H1                       AROS  0870
      TSTAR = TST1*FS(3)                                               AROS  0880
      IF (TSTAR.LE.225.0) VIS = 0.80382436E-9*TSTAR                    AROS  0890
      IF (TSTAR.GT.225.0) VIS = 2.27E-8*TSTAR**1.5/(TSTAR + 198.6)     AROS  0900
      ROSTAR = BS(1)*BS(3)/TSTAR                                       AROS  0910
      GO TO 170                                                        AROS  0920
C  REAL GAS - REFERENCE ENTHALPY                                       AROS  0930
  160 HW = ENTHAL(TW,PW)                                               AROS  0940
      HSTAR = A1*HW + A2*HAW + (1.-A1-A2)*H2                           AROS  0950
      TST1 = HSTAR/H1                                                  AROS  0960
      ROSTAR = ROW(HSTAR,PW)                                           AROS  0970
      VIS = ROMU(HSTAR,PW)/ROSTAR                                      AROS  0980
  170 FC = BS(1)/ROSTAR                                                AROS  0990
      FRX  =      ROSTAR*BS(5)/(BS(1)*VIS)                             AROS  1000
      RET  = FRX*BS(8)*ELLOC                                           AROS  1010
      IF (RET.LT.2540.0) GO TO (11,21), ITW                            AROS  1020
      CFTLOC = 0.088*(ALOG10(RET) - 2.3686)/(ALOG10(RET) - 1.5)**3/FC  AROS  1030
      QC = 0.5*(0.71**(-2./3.))*BS(1)*BS(7)*CFTLOC*(HAW - HW)          AROS  1040
      RETURN                                                           AROS  1050
      END                                                              AROS  1060
```




COMPUTED GO TO
IF THE VALUE TRANSFER
OF ITW TO STATEMENT
GO TO 60
COMPUTED GO TO
IF THE VALUE TRANSFER
OF ITW TO STATEMENT
GO TO 170


GO TO 3
YES
IS
RET.LT.2540.
NO
CFTLOC=0.088*(ALOG10(RET)-2.3686)/(ALOG10(RET)-1.5)**3/FC
QC=0.5*(0.71**(-2./3.))*BS(1)*BS(7)*CFTLOC*(HAW-HW)
RETURN

SYMBOLS USED IN SUBROUTINE QC

A	R	U	COEFFICIENT USED IN SPALDING-CHI METHOD	QC
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	QC
A1	R	U	COEFFICENT IN THE DEFINITION OF REFERENCE CONDITION	QC
A2	R	U	COEFFICIENT IN THE DEFINITION OF REFERENCE CONDITION	QC
B	R	U	COEFFICIENT USED IN SPALDING-CHI METHOD	QC
BS	R	C	FLOW CONDITION ARRAY BEHIND SHOCK OR EXPANSION	QC
C	R	U	COEFFICIENT USED IN SPALDING-CHI METHOD	QC
CASE	I	C	CASE NUMBER	QC
CFTLOC	R	C	LOCAL TURBULENT SKIN-FRICTION COEFFICIENT	QC
CKU	R	C	LAMINAR FLOW FLIGHT CONDITION CONSTANT	QC
ELLOC	R	C	REFERENCE LENGTH	QC
ERROR	I	C	ERROR FLAG	QC
FC	R	C	TURBULENT FLOW, SKIN FRICTION COMPRESSIBILITY FACTOR	QC
FRX	R	C	TURBULENT FLOW, REYNOLDS NUMBER COMPRESSIBILITY FACTOR	QC
FS	R	C	FREE-STREAM FLOW CONDITION ARRAY	QC
GCP	R	C	GAS SPECIFIC HEAT AT CONSTANT PRESSURE	QC
HAW	R	C	ADIABATIC-WALL ENTHALPY	QC
HSTAR	R	U	REFERENCE ENTHALPY	QC
HW	R	C	WALL ENTHALPY	QC
H1	R	C	FREE-STREAM ENTHALPY	QC
H2	R	C	LOCAL ENTHALPY	QC
IFLOW	I	C	LAMINAR (=1) OR TURBULENT (=2) FLOW FLAG	QC
IHW	I	C	WALL ENTHALPY FLAG	QC
IM	I	C	ELEMENT ROW NUMBER ARRAY	QC
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	QC
ITURB	I	C	TURBULENT FLOW FLAG	QC
ITW	I	C	IDEAL GAS (=1) OR REAL GAS (=2) FLAG	QC
L	I	C	NUMBER OF ELEMENTS	QC
LS	I	C	NUMBER OF ELEMENTS	QC
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	QC
NY2	R	C	ELEMENT DIRECTION COSINE ARRAY-Y	QC
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	QC
PAGE	I	C	PAGE NUMBER	QC
PW	R	U	PRESSURE	QC
QC	R	U	CONVECTIVE HEATING RATE	QC
RA	R	U	REYNOLDS ANALOGY FACTOR	QC

SYMBOLS USED IN SUBROUTINE QC

```
RET      R  C  TURBULENT FLOW REYNOLDS NUMBER AT REFERENCE CONDITION             QC
RUMURA   R  C  SQUARE-ROOT OF REFERENCE DENSITY-VISCOSITY RATIO                  QC
ROSTAR   R  U  DENSITY AT REFERENCE CONDITION                                    QC
TCT1     R  U  SUTHERLAND CONSTANT TO FREE-STREAM TEMPERATURE RATIO              QC
TITLE    R  C  TITLE                                                             QC
TSTAR    R  U  REFERENCE TEMPERATURE                                             QC
TST1     R  C  REFERENCE TO FREE-STREAM TEMPERATURE (OR ENTHALPY) RATIO          QC
TW       R  A  WALL TEMPERATURE                                                  QC
TW1      R  U  WALL TEMPERATURE                                                  QC
VIS      R  U  VICOSITY AT REFERENCE CONDITION                                   QC
VISRA    R  U  REFERENCE TO FREE-STREAM VISCOSITY RATIO                          QC
XCENT2   R  C  QUADRILATERAL ELEMENT CENTROID ARRAY-X                            QC
YCENT2   R  C  QUADRILATERAL ELEMENT CENTROID ARRAY-Y                            QC
ZCENT2   R  C  QUADRILATERAL ELEMENT CENTROID ARRAY-Z                            QC
```


QC

21. FUNCTION POLY (DECK AROT)

This routine generates an N-th order polynomial.

a. Algorithm

Polynomial is evaluated for the input order and coefficient array at a specified starting value.

b. Input/Output

None

c. Error

None

d. Subroutines Required

None

e. Argument List

(A, I, HX, N)

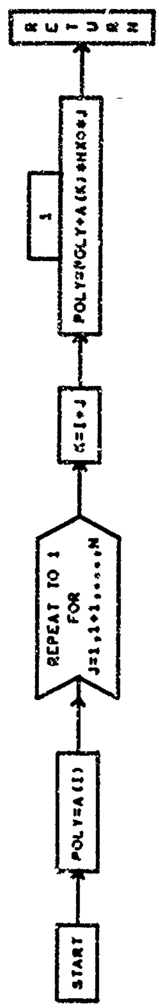
f. Length

444 bytes

DECK AROT

```
      FUNCTION POLY(A,I,HX,N)
      DIMENSION A(135)
C
C  THIS FUNCTION GENERATES AN N-TH ORDER POLYNOMIAL
C  IN HX WITH COEFFICIENTS A(K) STARTING WITH K=I.
C
      POLY = A(I)
      DO  1 J = 1,N
      K = I + J
    1 POLY = POLY + A(K)*HX**J
      RETURN
      END
```

FUNCTION POLY



SYMBOLS USED IN SUBROUTINE POLY

A	R	A	POLYNOMIAL COEFFICIENT ARRAY	POLY
HX	R	A	INDEPENDENT VARIABLE	POLY
I	I	A	INDEX NUMBER OF INITIAL COEFFICIENT	POLY
J	I	U	DO-LOOP INDEX	POLY
K	I	U	COEFFICIENT NUMBER	POLY
N	I	A	ORDER OF POLYNOMIAL	POLY
POLY	R	U	VALUE OF POLYNOMIAL	POLY

22. FUNCTION ROMU (DECK AROU)

This routine calculates various equilibrium air real gas properties. Has three entries; ROMU, ENTHAL, and ROW.

a. Algorithm

ROMU determines the density-viscosity product as a function of the input enthalpy and pressure.

ENTHAL determines the enthalpy as a function of the input temperature and pressure.

ROW calculates the density as a function of the input enthalpy and pressure.

b. Input/Output

None

c. Error

None

d. Subroutines Required

POLY

e. Argument List

ROMU (HS, P2), ENTHAL (TW, PW), ROW (HS, P2)

f. Length

2478 bytes

DECK AROU

```
      FUNCTION ROMU(HS,P2)
C
C
C  THIS FUNCTION HAS THREE ENTRIES.  THE FIRST, ROMU(HS,P2), CALCULATES
C  THE DENSITY-VISCOSITY PRODUCT FOR EQUILIBRIUM AIR AT THE INPUT
C  ENTHALPY (HS) AND PRESSURE (P2).
C              RANGE,  HS TO ABOUT 4.E+8 (FT/SEC)**2 AND P2 .GT. 1.E-4
C                      AND .LT. 10.  EXTRAPOLATION FOR P2 OUTSIDE THIS.
C  THE SECOND, ENTHAL(TW,PW), CALCULATES THE ENTHALPY CORRESPONDING
C  TO THE INPUT TEMPERATURE (TW) AND PRESSURE (PW).
C              RANGE,  TW TO ABOUT 8000.OR (PROGRAM HAS CUTOFF AT
C                      7000.OR)  AND PW SAME AS P2 ABOVE.
C  THE  THIRD, ROW(HS,P2), CALCULATES THE DENSITY (SLUGS/FT**3)
C  FOR EQUILIBRIUM AIR AT THE INPUT ENTHALPY (HS) AND PRESSURE (P2).
C              RANGE,  SAME AS FOR ROMU ABOVE.
C  ALL ENTRIES REQUIRE BLOCK DATA FOR COMMON/PROP/ AND FUNCTION
C  POLY.  THE PROPERTIES BASED ON AEDC TR-65-58 AND HANSENS VISCOSITY.
C  DETAILS OF THE PROPERTY FITS ARE GIVEN IN DOUGLAS REPORT LB-32706.
C**************D. N. SMYTH PROGRAM AUTHOR*******************************
C
C
C
      COMMON /PROP/ FH(135), FR(135)
      DIMENSION IH(3)
      DATA PREF/2117.36/
      H1 = HS*1.0E-8
C
C  DETERMINE ENTHALPY RANGE AT 10.0ATM
      IRM1 = 1
      IF (H1.LT.0.042) GO TO 1
      IRM1 = 10
      IF (H1.LT.0.1768) GO TO 1
      IRM1 = 46
      IF (H1.LT.0.650) GO TO 2
      IRM1 = 55
```

```
DECK AROU

      IF (H1.GT.1.10) IRM1 = 64
C
C   DETERMINE ENTHALPY RANGE AT 10.0**-4 ATM
    1 IF (H1.GT.0.1768) GO TO 2
      IRM2 = IRM1
      H1 = H1*10.0
      GO TO 3
    2 IRM2 = 19
      IF (H1.LT.0.580) GO TO 3
      IRM2 = 28
      IF (H1.GT.0.980) IRM2 = 37
    3 N = 6
      PBAR = P2/PREF
      PBARLG = ALOG10(PBAR)
      ROMU =-0.2*((PBARLG-1.0)*POLY(FR,IRM2,H1,N) - (PBARLG+4.0)*
     1           POLY(FR,IRM1,H1,N))
      ROMU = ROMU*PBAR*1.0E-9
      RETURN
C
C
C   DETERMINE ENTHALPY FOR GIVEN TEMPERATURE (MAXIMUM 7000.0R)
      ENTRY ENTHAL(TW,PW)
      N = 8
      IF (TW.GT.7000.0) TW = 7000.0
      THX = TW*1.0E-4
      J = 1
      IF (TW.GT.2700.) GO TO 10
      IH1 = 1
      ROMU = POLY(FH,IH1,THX,N)*1.0E+8
      RETURN
C
   10 PBAR = PW/PREF
      PBARLG = ALOG10(PBAR)
C
C   DETERMINE THREE PBARLG VALVES TO USE IN QUADRATIC INTERPOLATION
      IF (PBARLG.GT.-3.0) GO TO 20
```

DECK AROU

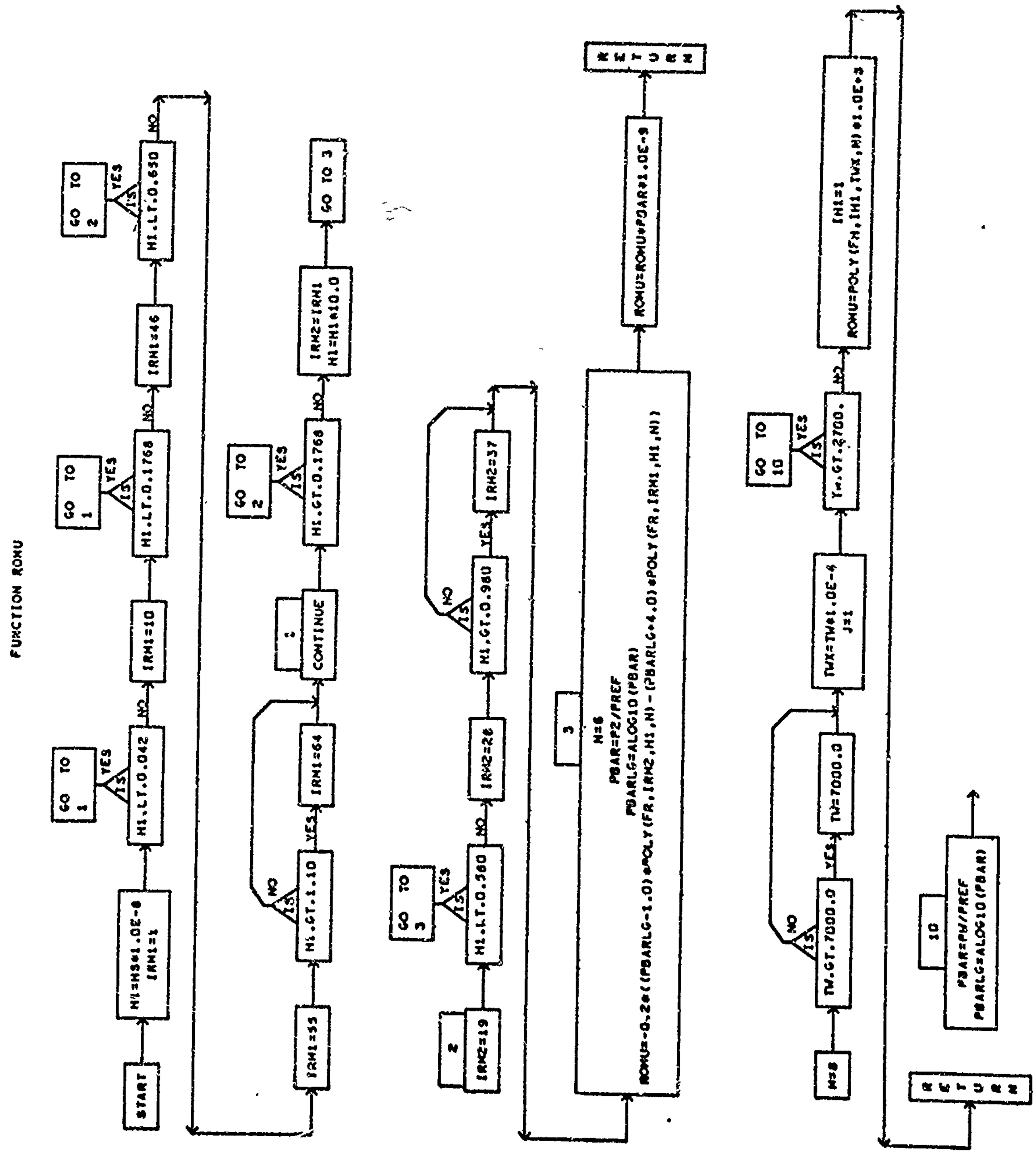
```
      PBAR1 = -4.0
      PBAR2 = -3.0
      PBAR3 = -2.0
      GO TO 12
   20 IF (PBARLG.GT.-2.0) GO TO 30
      PBAR1 = -3.0
      PBAR2 = -2.0
      PBAR3 = -1.0
      GO TO 14
   30 IF (PBARLG.GT.-1.0) GO TO 40
      PBAR1 = -2.0
      PBAR2 = -1.C
      PBAR3 =  0.0
      GO TO 16
   40 PBAR1 = -1.0
      PBAR2 =  0.0
      PBAR3 =  1.C
      GO TO 18
C
C   DETERMINE TEMP. RANGE AT PBARLG = -4
   12 IH(J) = 10
      IF (TW.LT.4200.) GO TO 13
      IH(J) = 19
      IF (TW.GT.5500.) IH(J) = 28
   13 J = J + 1
C
C   DETERMINE TEMP. RANGE AT PBARLG = -3
   14 IH(J) = 37
      IF (TW.LT.4540.) GO TC 15
      IH(J) = 46
      IF (TW.GT.6120.) IH(J) = 55
   15 J = J + 1
C
C   DETERMINE TEMP. RANGE AT PBARLG = -2
   16 IH(J) = 64
      IF (TW.LT.5220.) GO TO 17
```

DECK AROU

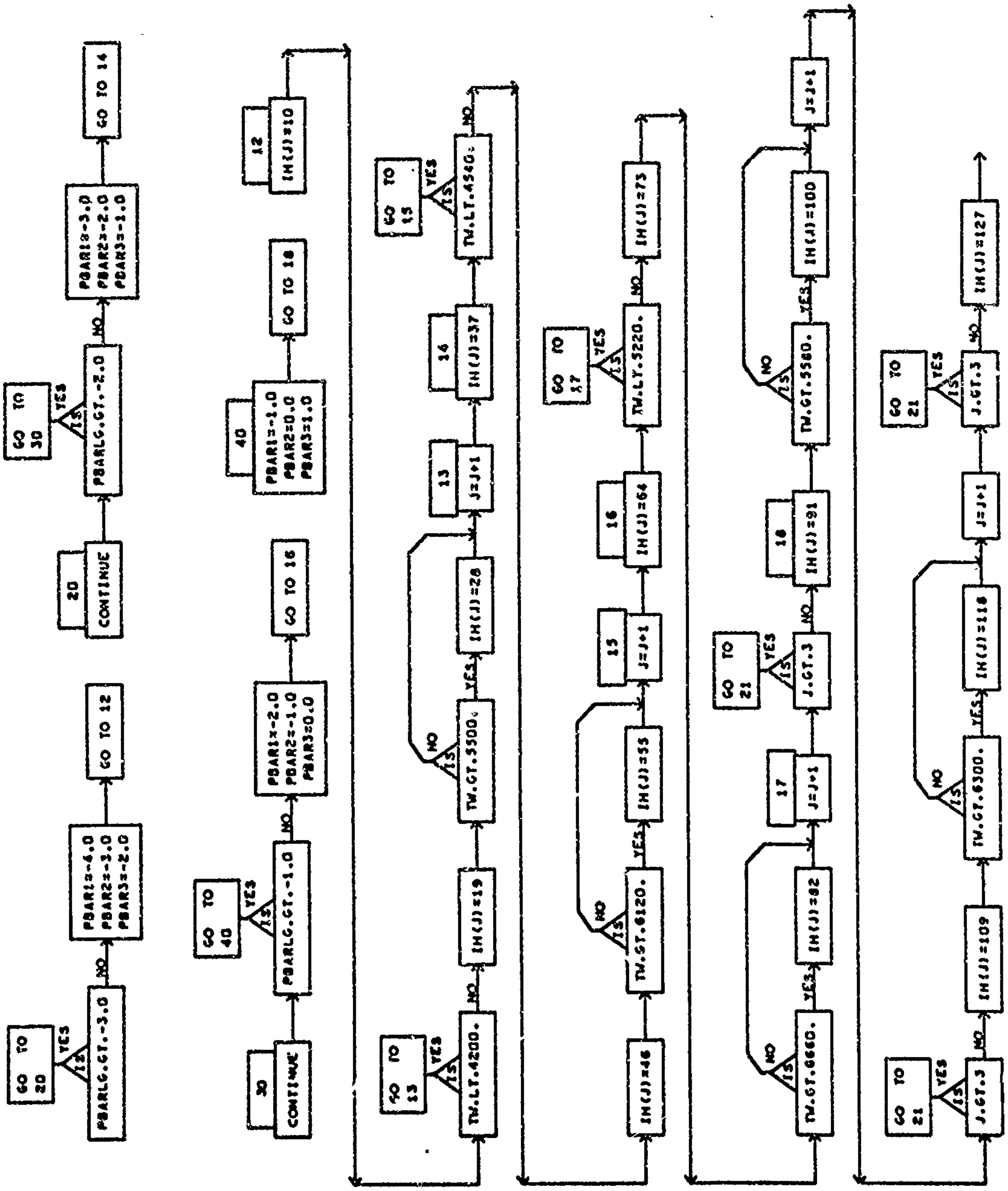
```
      IH(J) = 73
      IF (TW.GT.6660.) IH(J) = 82
   17 J = J + 1
      IF (J.GT.3) GO TO 21
C
C  DETERMINE TEMP. RANGE AT PBARLG = -1
   18 IH(J) = 91
      IF (TW.GT.5580.) IH(J) = 100
      J = J + 1
      IF (J.GT.3) GO TO 21
C  DETERMINE TEMP. RANGE AT PBARLG = 0
      IH(J) = 109
      IF (TW.GT.6300.) IH(J) = 118
      J = J + 1
      IF (J.GT.3) GO TO 21
C
C  DETERMINE TEMP. RANGE AT PBARLG = 1
      IH(J) = 127
C
C  CALCULATE ENTHALPY
   21 IH1 = IH(1)
      IH2 = IH(2)
      IH3 = IH(3)
      HC1 = (PBARLG - PBAR2)*(PBARLG - PBAR3)*0.5
      HC2 = (PBARLG - PBAR1)*(PBARLG - PBAR3)
      HC3 = (PBARLG - PBAR1)*(PBARLG - PBAR2)*0.5
      ROMU =(HC1*POLY(FH,IH1,TWX,N) - HC2*POLY(FH,IH2,TWX,N)
     1          + HC3*POLY(FH,IH3,TWX,N))*1.0E+8
      RETURN
C
C
C  DETERMINE DENSITY FOR GIVEN ENTHALPY AND PRESSURE.
      ENTRY  ROW(HS,P2)
      HI = HS*1.0E-8
      N = 9
C
```

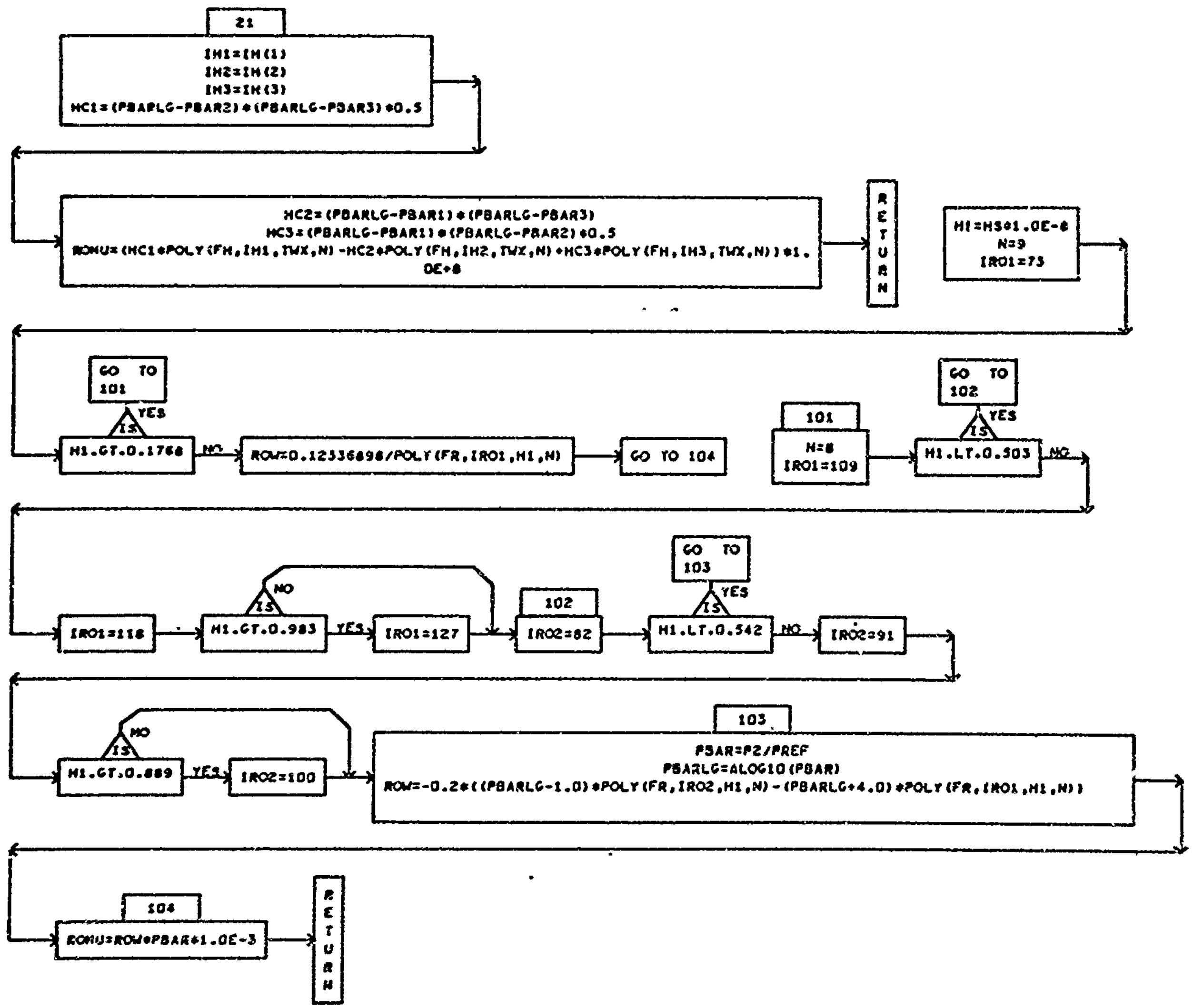
ROMU

```
DECK AROU

C DETERMINE ENTHALPY RANGE AT 10.0 ATM.
      IRO1  = 73
      IF (H1.GT.0.1768) GO TO 101
      ROW = 0.12336898/POLY(FR,IRO1, H1,N)
      GO TO 104
  101 N = 8
      IRO1  = 109
      IF (H1.LT.0.503) GO TO 102
      IRO1  = 118
      IF (H1.GT.0.983) IRO1  = 127
C
C  DETERMINE ENTHALPY RANGE AT 10.0**-4 ATM.
  102 IRO2 = 82
      IF (H1.LT.0.542) GO TO 103
      IRO2 = 91
      IF (H1.GT.0.889) IRO2 = 100
  103 PBAR = P2/PREF
      PBARLG = ALOG10(PBAR)
      ROW = -0.2*((PBARLG - 1.0)*POLY(FR,IRO2,H1,N) -
     1            (PBARLG + 4.0)*POLY(FR,IRO1,H1,N))
  104 ROMU = ROW*PBAR*1.0E-3
      RETURN
      END
```


FUNCTION ROMU
START
H1=HS*1.0E-8
IRM1=1
IS
H1.LT.0.042
YES
GO TO 1
NO
IRM1=10
H1.LT.0.1768
GO TO 1
IRM1=46
H1.LT.0.650
GO TO 2
IRM1=55
H1.GT.1.10
IRM1=64
1
CONTINUE
H1.GT.0.1768
GO TO 2
IRM2=IRM1
H1=H1*10.0
GO TO 3
2
IRM2=19
H1.LT.0.580
GO TO 3
IRM2=28
H1.GT.0.980
IRM2=37
3
N=6
PBAR=P2/PREF
PBARLG=ALOG10(PBAR)
ROMU=-0.2*((PBARLG-1.0)*POLY(FR,IRM2,H1,N)-(PBARLG+4.0)*POLY(FR,IRM1,H1,N))
ROMU=ROMU*PBAR*1.0E-9
RETURN
N=8
TW.GT.7000.0
TW=7000.0
TWX=TW*1.0E-4
J=1
TW.GT.2700.
GO TO 10
IH1=1
ROMU=POLY(FN,IH1,TWX,N)*1.0E+3
10
PBAR=PW/PREF
PBARLG=ALOG10(PBAR)
RETURN

ROMU





21
IH1=IH(1)
IH2=IH(2)
IH3=IH(3)
HC1=(PBARLG-PBAR2)*(PBARLG-PBAR3)*0.5
HC2=(PBARLG-PBAR1)*(PBARLG-PBAR3)
HC3=(PBARLG-PBAR1)*(PBARLG-PBAR2)*0.5
ROMU=(HC1*POLY(FH,IH1,TWX,N)-HC2*POLY(FH,IH2,TWX,N)+HC3*POLY(FH,IH3,TWX,N))*1.0E+8
RETURN
H1=HS*1.0E-8
N=9
IRO1=73
GO TO 101
YES
IS
H1.GT.0.1768
NO
ROW=0.12336898/POLY(FR,IRO1,H1,N)
GO TO 104
101
N=8
IRO1=109
GO TO 102
YES
IS
H1.LT.0.503
NO
IRO1=118
NO
IS
H1.GT.0.983
YES
IRO1=127
102
IRO2=82
GO TO 103
YES
IS
H1.LT.0.542
NO
IRO2=91
NO
IS
H1.GT.0.889
YES
IRO2=100
103
PSAR=P2/PREF
PBARLG=ALOG10(PBAR)
ROW=-0.2*((PBARLG-1.0)*POLY(FR,IRO2,H1,N)-(PBARLG+4.0)*POLY(FR,IRO1,H1,N))
104
ROMU=ROW*PBAR*1.0E-3
RETURN

POLY

ROMU

SYMBOLS USED IN SUBROUTINE ROMU

ENTHAL	R	U	ENTRY TO DETERMINE ENTHALPY	ROMU
FH	R	C	ENTHALPY ARRAY	ROMU
FR	R	C	DENSITY-VISCOSITY PRODUCT AND DENSITY ARRAYS	ROMU
HC1	R	U	FIRST ENTHALPY COEFFICIENT	ROMU
HC2	R	U	SECOND ENTHALPY COEFFICIENT	ROMU
HC3	R	U	THIRD ENTHALPY COEFFICIENT	ROMU
HS	R	A	ENTHALPY (FT/SEC)**2	ROMU
H1	R	U	REDUCED ENTHALPY (HS*1.0E-8)	ROMU
IH	I	D	ENTHALPY ARRAY INDEX	ROMU
IH1	I	U	ENTHALPY ARRAY INDEX AT FIRST PRESSURE	ROMU
IH2	I	U	ENTHALPY ARRAY INDEX AT SECOND PRESSURE	ROMU
IH3	I	U	ENTHALPY ARRAY INDEX AT THIRD PRESSURE	ROMU
IRM1	I	U	DENSITY-VISCOSITY ARRAY INDEX AT 10.0 ATM.	ROMU
IRM2	I	U	DENSITY-VISCOSITY ARRAY INDEX AT 10.0**-4 ATM.	ROMU
IRO1	I	U	DENSITY ARRAY INDEX AT 10.0 ATM.	ROMU
IRO2	I	U	DENSITY ARRAY INDEX AT 10.0**-4 ATM.	ROMU
J	I	U	ENTHALPY INDEX COUNTER	ROMU
N	I	U	ORDER OF POLYNOMIAL	ROMU
PBAR	R	U	PRESSURE RATIO (PW/PREF)	ROMU
PBARLG	R	U	LOG10 OF PBAR	ROMU
PBAR1	R	U	LOG10 OF FIRST REFERENCE PRESSURE RATIO	ROMU
PBAR2	R	U	LOG10 OF SECOND REFERENCE PRESSURE RATIO	ROMU
PBAR3	R	U	LOG10 OF THIRD REFERENCE PRESSURE RATIO	ROMU
PREF	R	U	REFERENCE PRESSURE (2117.36 LB/SQ.FT.)	ROMU
PW	R	A	LOCAL PRESSURE (LB/SQ.FT.)	ROMU
P2	R	A	LOCAL PRESSURE (LB/SQ.FT.)	ROMU
ROMU	R	U	DENSITY-VISCOSITY PRODUCT	ROMU
ROW	R	U	ENTRY TO DETERMINE DENSITY (ALSO DENSITY PARAMETER)	ROMU
TW	R	A	TEMPERATURE (RANKINE)	ROMU
TWX	R	U	REDUCED TEMPERATURE (TW*1.0E-4)	ROMU

23. BLOCK DATA (DECK AROV)

This routine initializes data arrays into common required in calculating equilibrium air real gas properties.

a. Algorithm

Data arrays are initialized at time of compilation into labelled common /PROP/.

b. Input/Output

None

c. Error

None

d. Subroutines Required

None

e. Argument List

None

f. Length

1080 bytes

DECK AROV

```
      BLOCK DATA
C
C
C  THIS SUBROUTINE INITIALIZES INTO CCMMON/PROP/  THE COEFFICIENT
C  ARRAYS REQUIRED BY FUNCTION ROMU (ARO3)  TO DETERMINE THE
C  REAL EQUILIBRIUM AIR PROPERTIES.
C
C
C
         DIMENSION FH(135),FR(135),FH1(108),FH2(27),FR1(72),FR2(63)
      COMMON /PROP/ FH,FR
      EQUIVALENCE  (FH(1),FH1(1)),(FH(109),FH2(1)),
     1              (FR(1),FR1(1)), (FR(73),FR2(1))
C
C
      DATA                FH1  / 0.0, .60181771, -.22228717, 2.5953429,
     1 -2.7853922, -9.7092052, 23.289510, -12.199244, -2.5817427,
     2 -.12935540, -2.9966848, 48.010840, -184.77242, 249.48974,
     3 -51.244932, 0.0, 0.0, 0.0,   -6.5939838, 19.010757, 19.992531,
     4 -39.975681, -112.89466, 156.16378, 0.0, 0.0, 0.0,   2.2022949,
     5 -14.101200, 23.004509, 52.890407, -165.08756, 113.04871, 0.0,
     6 0.0, 0.0,    -.20293060, .71240021, 11.699213, -47.629621,
     7 42.913642, 30.631199, 0.0, 0.0, 0.0,    1.8692814, -22.097879,
     8 56.153394, -16.042275, 36.738951, -263.49237, 239.87072, 0.0,
     9 0.0,    -.15686170, 1.1830883, -2.0213376, 19.029276, -26.810587,
     A -9.9053574, 24.334188, 0.0 , 0.0,    .92262030, -8.5032999,
     B 20.452311, 45.587738, -119.90206, -509.58200, 1721.3813,
     C -1337.6228, 0.0,    -2.6782862, 1.2865940, 22.104893, -12.395676,
     D -26.121859, -12.897718, 38.794573, 0.0, 0.0,    -1.6135321,
     E 5.3650713, 2.4832548, -9.7412082, 1.2468173, .39646271, 4.5573527
     F,0.0, 0.0,    .44553169, -3.0252481, 2.6196596, 50.144314,
     G -151.34585, 133.67463, -154.12089, 535.69899, -494.43566,
     H -3.0416265, 12.560193, -20.33612, 17.381598, 23.843668,
     I -64.498613, 36.134646, 0.0, 0.0/
C
```

```
DECK ARDV
C                                                                       ARDV 0360
      DATA               FH2  / .1199213, -.65116698, 3.8928011,        ARDV 0370
     1 -.45545770, -5.3407127, -21.592914, 33.774256, 58.270657,        ARDV 0380
     2 -78.177197,    .34683226, -.79201527, -.79886544, 5.2064374,     ARDV 0390
     3 -1.2818503, 4.3016173, -6.9417639, 0.0, 0.0,   .11764422,        ARDV 0400
     4 -.62580124, 3.5461409, 4.8925829, -39.857784, 71.680475,         ARDV 0410
     5 -48.794024, 9.9564413, 0.0 /,                                    ARDV 0420
     6              FR1 / 1.0490322, -.31228101,.90103854E-1,-6.7117799, ARDV 0430
     7 18.184363, -13.551560, 0.0, 0.0, 0.0,   1.1329366, -.94702472,   ARDV 0440
     8 .55849455,-.92146834E-1,-.50705935E-1,.17519852E-1,0.0,0.0,0.0,  ARDV 0450
     9 .88808527, -3.9763502, 14.251132, -27.521840, 26.791510,         ARDV 0460
     A -10.436079, 0.0, 0.0, 0.0,    -.14569908, 3.0171600, -5.9999473, ARDV 0470
     B 4.6748398, -1.2742339, 0.0, 0.0, 0.0, 0.0, .37102156, -.13586302, ARDV 0480
     C.44261289E-1,-.78199161E-2,.52479786E-3,0.0, 0.0, 0.0, 0.0,       ARDV 0490
     D .77497574, -2.1673001, 4.0652322, -2.7726303, -1.1814257,        ARDV 0500
     E 1.6918670, 0.0, 0.0, 0.0,    .52865913, -.58894771, .58988212,   ARDV 0510
     F -.42483488, .19163722,-.38356068E-1,0.0, 0.0, 0.0,    .48598990, ARDV 0520
     G -.41125640, .25298705, -.82910932E-1,.13654598E-1, -.89113877E-3, ARDV 0530
     H 0.0, 0.0, 0.0/                                                   ARDV 0540
C                                                                       ARDV 0550
C                                                                       ARDV 0560
      DATA               FR2  / 0.0, 1.6627798, 1.1792685,              ARDV 0570
     1 -28.017959, 155.77436, -439.43620, 709.71808, -671.23004,        ARDV 0580
     2 317.22690,    1.5380637, -11.161238, 32.383428, 2.5847158,       ARDV 0590
     3 -196.69668, 370.82431, -218.12898, 0.0, 0.0,   1.3507331,        ARDV 0600
     4 -5.0193304, 6.0404391, 4.1730680, -13.136917, 6.7913235, 0.0,    ARDV 0610
     5 0.0, 0.0,   .42795709, -.56949512, .40404761, -.11638753,        ARDV 0620
     6 -.74634946E-2, .12505798E-1,-.27304295E-2, .19434979E-3, 0.0,    ARDV 0630
     71.3644197, -9.5027751, 34.791129, -68.553220, 69.547355,          ARDV 0640
     8 -28.523442, 0.0, 0.0, 0.0,   .48102262, -.91625506, .62421014,   ARDV 0650
     9 .57730033, -1.0682441, .42044702, 0.0, 0.0, 0.0,   .42220834,    ARDV 0660
     A -.61202419, .44184292, -.15420936,  .18083084E-1,.36418043E-2,   ARDV 0670
     B   -.12611833E-2, .10222056E-3, 0.0 /                             ARDV 0680
C                                                                       ARDV 0690
C                                                                       ARDV 0700
      END                                                               ARDV 0710
```


DATA

SYMBOLS USED IN SUBROUTINE DATA

```
FH       R  C  ENTHALPY ARRAY                                        DATA
FH1      R  D  FIRST 108 ELEMENTS OF ENTHALPY ARRAY                  DATA
FH2      R  D  FINAL 27 ELEMENTS OF ENTHALPY ARRAY                   DATA
FR       R  C  DENSITY-VISCOSITY PRODUCT AND DENSITY ARRAYS          DATA
FR1      R  D  DENSITY-VISCOSITY PRODUCT ARRAY                       DATA
FR2      R  D  DENSITY ARRAY                                         DATA
```

24. SUBROUTINE ATMOS (DECK AROW)

This routine calculates the atmospheric properties using the 1962 U. S. Atmosphere.

a. Algorithm

Set up arrays and constant values. Calculate atmospheric properties assuming an inverse square gravitational field.

b. Input/Output

c. Error

None

d. Subroutines Required

None

e. Argument List

(A3, A8, A4, A1, A6)

f. Length

1792 bytes

DECK AROW

```
      SUBROUTINE ATMOS (A3,A8,A4,A1,A6)

C     THIS ROUTINE CALCULATES ATMOSPHERIC PROPERTIES OF THE
C     US STANDARD ATMOSPHERE,1962,ASSUMING AN INVERSE SQUARE
C     GRAVITATIONAL FIELD. THIS ASSUMPTION YIELDS DATA THAT
C     AGREES WITH THE COESA DOCUMENT WITHIN 1 PER CENT AT
C     ALL ALTITUDES UP TO 700 KILOMETERS (2296588 FEET). THE
C     DATA IS ARRANGED IN THE ATMOSPHERE ARRAY, A, AS
C     FOLLOWS
C     A(1)  = CS, SPEED OF SOUND, FT/SEC
C     A(2)  = (1/CS)(DCS/DZ), SOUND DERIVATIVE, 1/FT
C     A(3)  = Z, GEOMETRIC ALTITUDE, FT (GIVEN)
C     A(4)  = P, PRESSURE, LB/FT2
C     A(5)  = DP/DZ, PRESSURE DERIVATIVE, LB/FT3
C     A(6)  = RHO, DENSITY, SLUGS/FT3
C     A(7)  = (1/RHO)(DRHO/DZ), DENSITY DERIVATIVE, 1/FT
C     A(8)  = T, TEMPERATURE, DEG RANKINE
C     A(9)  = DT/DZ, TEMPERATURE DERIVATIVE, DEG RANKINE/FT

C     VARIOUS CONSTANTS USED
C      EARTH RADIUS                  = 20890855 FT
C      SPECIFIC HEAT RATIO FOR AIR   = 1.4
C     SEA LEVEL VALUES
C      GRAVITATIONAL ACCELERATION    = 32.1740484 FT/SEC2
C      MOLECULAR WEIGHT              = 28.9644
C      GO*MO/R*                      = 0.018743418 DEG RANK/FT

      DIMENSION  A( 9),HG(10),ZM(14),WM(14),TM(23),PM(22)

C     SET ARRAYS AND CONSTANT VALUES
      DATA GO,WMO,RO,GMRS/32.1740484,28.9644,20890855.0,
     1     0.018743418/,HG/-16404.,0.0
     2     ,36089.,65617.,104987.,154199.,170604.,200131.,
     3     259186.,291160./,ZM/295276.,328084.,
     4     360892.,393701.,492126.,524934.,557743.,623360.,
```

```
DECK AROW

      5      754593.,984252.,1312336.,1640420.,1968504.,
      6      2296588./,WM/28.9644,28.88,28.56,
      7      28.07,26.92,26.66,26.4,25.85,24.7,22.66,19.94,
      8      17.94,16.84,16.17/

       DATA  TM/577.17,518.67,389.97,389.97,411.57
      1      ,487.17,487.17,454.77,325.17,325.17,379.17,469.17
      2      ,649.17,1729.17,1999.17,2179.17,2431.17,2791.17
      3      ,3295.17,3889.17,4357.17,4663.17,4861.17/,PM/
      4      3711.0839,2116.2165,472.67563,114.34314,
      5      18.128355,2.3162178,1.2321972,3.8030279E-01,
      6      2.1671352E-02,3.4313478E-03,6.2773411E-04,1.53490
      7      91E-04,5.2624212E-05,1.0561806E-05,7.7083076E-06,
      8      5.8267151E-06,3.9159854E-06,1.4520255E-06,3.92905
      9      63E-07,8.4030242E-08,2.2835256E-08,7.1875452E-09/

       A(3) = A3

C      CALCULATE G, Z,  ND CHECK
     2 Z = A(3)
       G = G0*(RO/(RO+Z))**2
       IF (Z .GT. 295276.0) GO TO 6

C      TMS LINEAR WITH GEOPOTENTIAL. CALCULATE H AND SEARCH
       H      = RO*Z/(RO+Z)
       DO 3 I = 2,10
       J      = I - 1
       IF (HG(I) .GE. H) GO TO 4
     3 CONTINUE

C      CALCULATE TMS SLOPE,TMS, AND SET MOL WT STUFF
     4 ELH    = (TM(J+1) - TM(J))/(HG(J+1) - HG(J))
       TMS    = TM(J) +  ELH*(H - HG(J))
       ELZ    = ELH*G/G0
       DMDZ   = 0.0
       EM     = WM0
```

DECK AROW

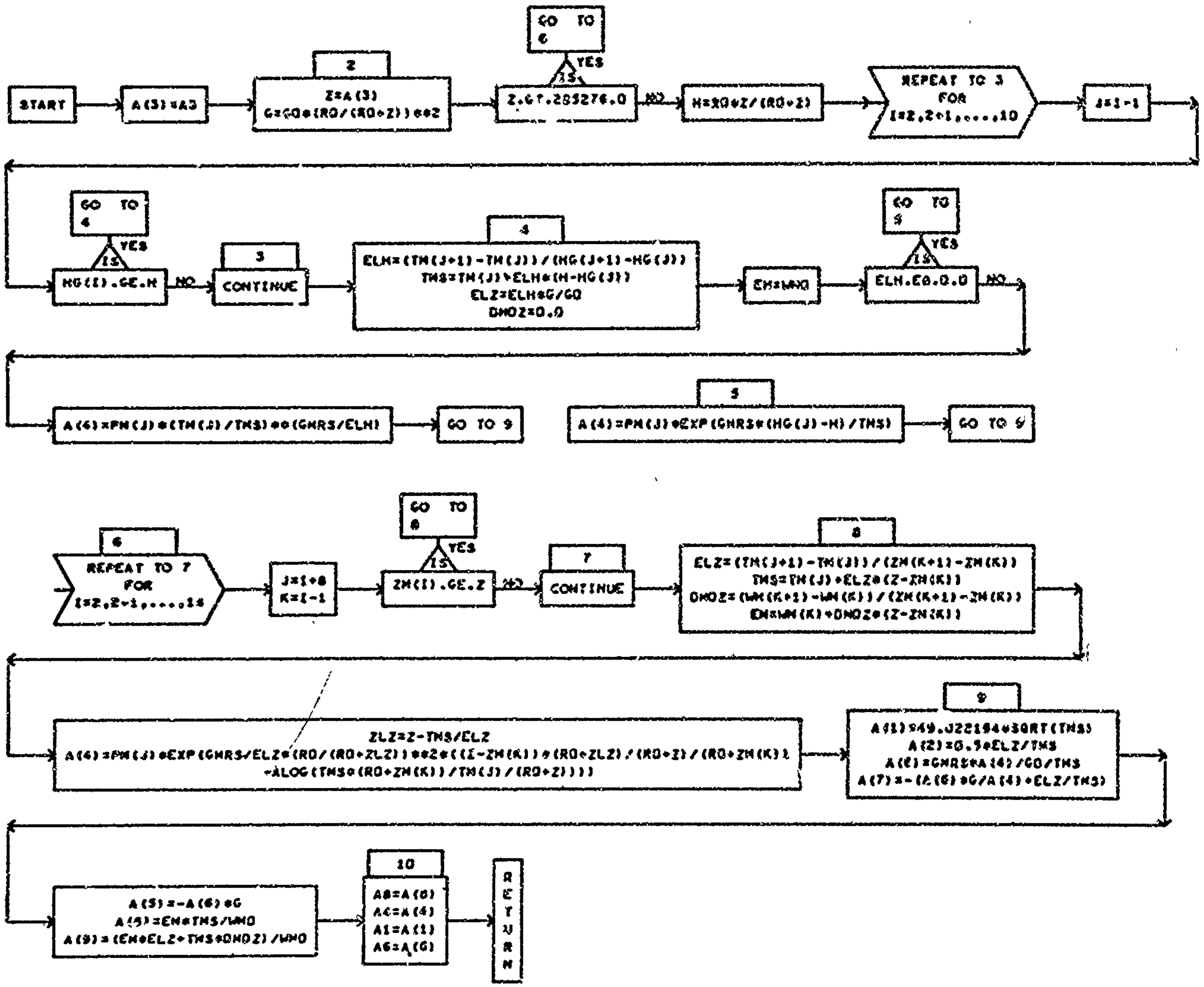
```
C     CHECK TMS SLOPE AND CALCULATE PRESSURE
      IF (ELH .EQ. 0.0) GO TO 5

C     NON - ZERO SLOPE PRESSURE EQUATION
      A(4) = PM(J)*(TM(J)/TMS)**(GMRS/ELH)
      GO TO 9

C     ZERO SLOPE PRESSURE EQUATION
    5 A(4) = PM(J)*EXP(GMRS*(HG(J)-H)/TMS)
      GO TO 9

C     TMS LINEAR WITH Z. SEARCH MATRIX
    6 DO 7 I = 2,14
      J      = I + 8
      K      = I - 1
      IF (ZM(I) .GE. Z) GO TO 8
    7 CONTINUE

C     CALCULATE TMS, SLOPE, AND STUFF
    8 ELZ  = (TM(J+1) - TM(J))/(ZM(K+1) - ZM(K))
      TMS  = TM(J) + ELZ*(Z - ZM(K))
      DMDZ = (WM(K+1) - WM(K))/(ZM(K+1) - ZM(K))
      EM   = WM(K) + DMDZ*(Z - ZM(K))
      ZLZ  = Z - TMS/ELZ

C     PRESSURE EQUATION FOR TMS LINEAR WITH Z
      A(4) = PM(J)*EXP(GMRS/ELZ*(RO/(RO+ZLZ))**2*((Z-ZM(K))*
     1       (RO+ZLZ)/(RO+Z)/(RO+ZM(K)) - ALOG(TMS*(RO+ZM(K)
     2       )/TM(J)/(RO+Z))))

C     CALCULATE SOUND SPEED AND DERIVATIVE
    9 A(1) = 49.022164*SQRT(TMS)
      A(2) = 0.5*ELZ/TMS

C     CALCULATE DENSITY, DERIVATIVE, AND PRESSURE DERIVATIVE
```

```
DECK AROW

      A(6) = GMRS*A(4)/GO/TMS
      A(7) = - (A(6)*G/A(4) + ELZ/TMS)
      A(5) = - A(6)*G

C     CALCULATE TEMPERATURE, DERIVATIVE, AND LEAVE
      A(8) = EM*TMS/WMO
      A(9) = (EM*ELZ + TMS*DMDZ)/WMO
   10 A8 = A(8)
      A4 = A(4)
      A1 = A(1)
      A6 = A(6)
      RETURN
      END
```

ATMOS

[redacted]

SUBROUTINE ATMOS



```
SYMBOLS USED IN SUBROUTINE ATMOS

A      R  D  MATRIX OF ATMOSPHERIC PROPERTIES                                    ATMOS
A1     R  A  ATMOSPHERIC SPEED OF SOUND, FEET/SECOND                             ATMOS
A3     R  A  GEOMETRIC ALTITUDE, FEET                                            ATMOS
A4     R  A  ATMOSPHERIC PRESSURE, POUNDS PER SQUARE FOOT                        ATMOS
A6     R  A  ATMOSPHERIC DENSITY, SLUGS PER CUBIC FOOT                           ATMOS
A8     R  A  ATMOSPHERIC TEMPERATURE, DEGREE RANKINE                             ATMOS
DMDZ   R  U  DERIVATIVE OF MOLECULAR WEIGHT OF AIR                               ATMOS
ELH    R  U  MOLECULAR SCALE TEMPERATURE DERIVATIVE, DEGREE RANKINE/FOOT         ATMOS
ELZ    R  U  MOLECULAR SCALE TEMPERATURE DERIVATIVE, DEGREE RANKINE/FOOT         ATMOS
EM     R  U  MOLECULAR WEIGHT OF AIR                                             ATMOS
G      R  U  GRAVITATIONAL ACCELERATION, FEET PER SECOND SQUARED                 ATMOS
GMRS   R  U  COMBINATION OF GEODETIC AND GAS CONSTANTS, DEG RANKINE/FOOT         ATMOS
GO     R  U  GRAVITATIONAL ACCELERATION AT SEA LEVEL, FT/SEC SQUARED             ATMOS
H      R  U  GEOPOTENTIAL ALTITUDE, FEET                                         ATMOS
HG     R  D  MATRIX OF GEOPOTENTIAL ALTITUDES, FT                                ATMOS
I      I  U  DO LOOP INDEX WHEN DETERMINING APPROPRIATE ATMOSPHERE LAYER         ATMOS
J      I  U  COUNTER IN VARIOUS DO LOOPS                                         ATMOS
K      I  U  COUNTER IN DO LOOP DETERMINING APPROPRIATE ATMOSPHERE LAYER         ATMOS
PM     R  D  MATRIX OF ATMOSPHERIC PRESSURES                                     ATMOS
RO     R  U  EARTH RADIUS = 20890855 FEET                                        ATMOS
TM     R  D  MATRIX OF MOLECULAR SCALE TEMPERATURES, DEG RANKINE                 ATMOS
TMS    R  U  MOLECULAR SCALE TEMPERATURE, DEGREE RANKINE                         ATMOS
WM     R  D  MATRIX OF MOLECULAR WEIGHTS OF AIR                                  ATMOS
WMO    R  U  MOLECULAR WEIGHT OF AIR AT SEA LEVEL = 28.9644                      ATMOS
Z      R  U  GEOMETRIC ALTITUDE, FEET                                            ATMOS
ZLZ    R  U  INTERIM CALCULATION FOR PRESSURE EQUATION.                          ATMOS
ZM     R  D  MATRIX OF GEOMETRIC ALTITUDES, FEET, ABOVE 245276 FEET              ATMOS
```

ATMOS

SYMBOLS USED IN SUBROUTINE ATMOS

A	R	D	MATRIX OF ATMOSPHERIC PROPERTIES	ATMOS
A1	R	A	ATMOSPHERIC SPEED OF SOUND, FEET/SECOND	ATMOS
A3	R	A	GEOMETRIC ALTITUDE, FEET	ATMOS
A4	R	A	ATMOSPHERIC PRESSURE, POUNDS PER SQUARE FOOT	ATMOS
A6	R	A	ATMOSPHERIC DENSITY, SLUGS PER CUBIC FOOT	ATMOS
A8	R	A	ATMOSPHERIC TEMPERATURE, DEGREE RANKINE	ATMOS
DMDZ	R	U	DERIVATIVE OF MOLECULAR WEIGHT OF AIR	ATMOS
ELH	R	U	MOLECULAR SCALE TEMPERATURE DERIVATIVE, DEGREE RANKINE/FOOT	ATMOS
ELZ	R	U	MOLECULAR SCALE TEMPERATURE DERIVATIVE, DEGREE RANKINE/FOOT	ATMOS
EM	R	U	MOLECULAR WEIGHT OF AIR	ATMOS
G	R	U	GRAVITATIONAL ACCELERATION, FEET PER SECOND SQUARED	ATMOS
GMRS	R	U	COMBINATION OF GEODETIC AND GAS CONSTANTS, DEG RANKINE/FOOT	ATMOS
GO	R	U	GRAVITATIONAL ACCELERATION AT SEA LEVEL, FT/SEC SQUARED	ATMOS
H	R	U	GEOPOTENTIAL ALTITUDE, FEET	ATMOS
HG	R	D	MATRIX OF GEOPOTENTIAL ALTITUDES, FT	ATMOS
I	I	U	DO LOOP INDEX WHEN DETERMINING APPROPRIATE ATMOSPHERE LAYER	ATMOS
J	I	U	COUNTER IN VARIOUS DO LOOPS	ATMOS
K	I	U	COUNTER IN DO LOOP DETERMINING APPROPRIATE ATMOSPHERE LAYER	ATMOS
PM	R	D	MATRIX OF ATMOSPHERIC PRESSURES	ATMOS
RO	R	U	EARTH RADIUS = 20890855 FEET	ATMOS
TM	R	D	MATRIX OF MOLECULAR SCALE TEMPERATURES, DEG RANKINE	ATMOS
TMS	R	U	MOLECULAR SCALE TEMPERATURE, DEGREE RANKINE	ATMOS
WM	R	D	MATRIX OF MOLECULAR WEIGHTS OF AIR	ATMOS
WMO	R	U	MOLECULAR WEIGHT OF AIR AT SEA LEVEL = 28.9644	ATMOS
Z	R	U	GEOMETRIC ALTITUDE, FEET	ATMOS
ZLZ	R	U	INTERIM CALCULATION FOR PRESSURE EQUATION.	ATMOS
ZM	R	D	MATRIX OF GEOMETRIC ALTITUDES, FEET, ABOVE 245276 FEET	ATMOS

25. SUBROUTINE PLUNGE (DECK AROX)

This routine calculates $C_{m_{\dot{\alpha}}}$ and $C_{Y_{\dot{\beta}}}$ for wings, bodies, and tails.

a. Algorithm

Read the Plunge Derivative Control Card (Type 14). Check correctness of input data flags. Read in plunge derivative data (Card Types 15, 16, 17, 18, 19, 20, 21) as required. Select proper calculation method, determine KBW if required, and calculate final derivatives.

b. Input/Output

Card Types 14, 15, 16, 17, 18, 19, 20, and 21

c. Error

An error condition occurs if the input flags are wrong, or if the card types are in error.

d. Subroutines Required

ELP1

e. Argument List

(IDERIV, CMA, CYB, CMADT, CYBDT)

f. Length

5540 bytes

DECK AROX

```
      SUBROUTINE PLUNGE (IDERIV,CMA,CYB,CMADT,CYBDT)
C
C***********************************************************************
C**** THIS ROUTINE COMPUTES C-SUB-M-SUB-ALPHA-DOT AND C-SUB-Y-SUB-BETA-
C**** DOT FOR WINGS, BODIES, OR TAILS
C***********************************************************************
C                    THIS SUBROUTINE WRITTEN BY J. LUNDRY
C
C
C     THE MAJOR CONTROL FLAGS FOR THIS SUBROUTINE ARE GIVEN BELOW
C         IPART = 1    CALCULATION FOR WING
C         IPART = 2    CALCULATION FOR BODY
C         IPART = 3    CALCULATION FOR TAIL
C
C         ICALC = 1    CALCULATION OF C-SUB-M-SUB-ALPHA-DOT
C         ICALC = 2    CALCULATION OF C-SUB-Y-SUB-BETA-DOT
C
C         K    IS FLAG CONTROLLING EQUATION FOR KBW.
C         COEF IS THE DESIRED COEFFICIENT DERIVATIVE
C
      DIMENSION  TITLE(15)
      COMMON  CASE,TITLE,PAGE,ERROR
      REAL KWB , KBW , M , LAMBDA, LENGTH
      INTEGER  TYPE,ERROR,CASE,PAGE
C
C     ARITHMETIC STATEMENT FUNCTIONS ARC-COSH, ARC-TANH
      ACOSH(X) = ALOG (X + SQRT(X**2 - 1.0))
      ATANH(X) = 0.5 * ALOG ( (1.0 + X) / (1.0 - X) )
C
      ERROR = 0
      IF (IDERIV .EQ. 5) ICALC = 1
      IF (IDERIV .EQ. 6) ICALC = 2
      PI = 3.1415926
C
C
```

DECK AROX

```
C ***** READ IN INPUT DATA ********************************************
C
C
      READ (5,101) IPART , TYPE
  101 FORMAT (I1, 69X, I2)
      IF (TYPE .NE. 14) GO TO 1000
C
C     CHECK IPART AND ICALC
C
      IF (IPART .GT. 0 .AND. IPART .LE. 3 .AND.
     A    ICALC .GT. 0 .AND. ICALC .LE. 2
     B   ) GO TO 61
      WRITE (6,102) IPART , ICALC
  102 FORMAT (1H , 73H**** BASIC INPUT FLAG ERRORS - SUBROUTINE PLUNGE
     A   IPART AND ICALC ARE  , 2I6 //22H **** TO ERR IS HUMAN  //)
      GO TO 1000
   61 IF (IPART .EQ. 2) GO TO (63,64) , ICALC
      IF (ICALC .EQ. 1) GO TO 62
      COEF = 0.0
      GO TO 90
   62 CONTINUE
      IF (IPART .EQ. 3) GO TO 105
C
C     READ WING DATA FOR C-SUB-M-SUB-ALPHA-DOT
C
      READ (5,103) AR , LAMBDA , M , SWBYS , CWBYC , S , K , TYPE
  103 FORMAT (6F10.0 , I1, 9X, I2)
      IF (TYPE .NE. 15) GO TO 1000
      READ (5,104) CR , R , BETA , CLALW , CMWPR , KBW , TYPE
  104 FORMAT (6F10.0 , 10X, I2)
      IF (TYPE .NE. 16) GO TO 1000
      GO TO 107
  105 CONTINUE
C
C     READ TAIL DATA FOR C-SUB-M-SUB-ALPHA-DOT
C
```

PLUNGE

DECK AROX

```
      READ (5,103) AR , LAMBDA , M , SWBYS , GAMMAT , S , K , TYPE        AROX 0720
      IF (TYPE .NE. 17) GO TO 1000                                        AROX 0730
      READ (5,104) CR , R , BETA , CLALW , UPWASH , XBTBYC , TYPE         AROX 0740
      IF (TYPE .NE. 18) GO TO 1000                                        AROX 0750
      READ (5,106) KBW , Q, TYPE                                          AROX 0760
  106 FORMAT (2F10.0, 50X, I2)                                            AROX 0770
      IF (TYPE .NE. 19) GO TO 1000                                        AROX 0780
  107 CONTINUE                                                            AROX 0790
C                                                                         AROX 0800
C ********************************************************************    AROX 0810
C ***** WING OR TAIL CONTRIBUTION                                         AROX 0820
C     THE SLENDER BODY THEORY IS USED TO DETERMINE KWB.  THE EXPRESSION   AROX 0830
C     IS EQ. (14) OF NACA REPORT 1307.                                    AROX 0840
C                                                                         AROX 0850
      D = 2.0 * R                                                         AROX 0860
      RS = R / S                                                          AROX 0870
      SR = S / R                                                          AROX 0880
      KWB = 2.0 / PI * (    (1.0 + RS ** 4) * (0.5 * ATAN (0.5 * (SR - RS AROX 0890
     A       )) + PI / 4.0) - RS ** 2 * (SR - RS + 2.0 * ATAN (RS))    )  AROX 0900
     B       / (1.0 - RS) ** 2                                            AROX 0910
C                                                                         AROX 0920
C ****CALCULATE KBW ************************************************      AROX 0930
C     SEVERAL EXPRESSIONS ARE TAKEN FROM NACA REPORT 1307 FOR KBW.        AROX 0940
C     THIS ROUTINE GIVES THE FOLLOWING OPTIONS AS A FUNCTION OF THE       AROX 0950
C     INTEGER ITYPE (IF EQ. (22) OF REPORT 1307 IS NOT SATISFIED,         AROX 0960
C     SLENDER BODY THEORY IS AUTOMATICALLY USED UNLESS KBW IS INPUT       AROX 0970
C     BY THE USER).                                                       AROX 0980
C                                                                         AROX 0990
C ITYPE  REFERENCE FOR KBW                                                AROX 1000
C   0      USER LOADS A VALUE OF KBW                                      AROX 1010
C   1      NACA REPT. 1307 EQ. (21)  (SLENDER BODY THEORY)                AROX 1020
C   2      NACA REPT. 1307 EQ. (24)     SUPERSONIC LEADING EDGE           AROX 1030
C                              (26)     SUBSONIC LEADING EDGE             AROX 1040
C              HALF-PLANFORM IS A TRAPEZOID.  WING/TAIL ON LONG BODY.     AROX 1050
C   3      NACA REPT. 1307 EQ. (27)                                       AROX 1060
C              RECTANGULAR PLANFORM.  WING/TAIL ON LONG BODY.             AROX 1070
```

```
DECK AROX
C    4      NACA REPT. 1307 EQ. (28)    SUBSONIC LEADING EDGE              AROX  1080
C                           EQ. (29)    SUPERSONIC LEADING EDGE            AROX  1090
C               TRIANGULAR PLANFORM.  WING/TAIL ON LONG BODY.              AROX  1100
C    5      NACA REPT. 1307 EQ. (30)    SUPERSONIC LEADING EDGE            AROX  1110
C                           EQ. (31)    SUBSONIC LEADING EDGE              AROX  1120
C               HALF-PLANFORM IS A TRAPEZOID.  NO AFTER-BODY FOR WING.     AROX  1130
      ITYPE = K                                                            AROX  1140
C                                                                          AROX  1150
C     TEST ITYPE FOR VALID RANGE                                           AROX  1160
C                                                                          AROX  1170
      IF (ITYPE .GE. 0 .AND. ITYPE .LE. 5) GO TO 72                        AROX  1180
      WRITE (6,71) ITYPE                                                   AROX  1190
   71 FORMAT(1H ,107H**** SUBROUTINE PLUNGE - THE FLAG ITYPE (WHICH CONT  AROX  1200
     2ROLS EQUATION USED TO CALCULATE KBW) IS INCORRECT AND =  ,I7 //)    AROX  1210
      GO TO 1000                                                           AROX  1220
   72 CONTINUE                                                             AROX  1230
C                                                                          AROX  1240
C                                                                          AROX  1250
C     TEST FOR EQ. (22) FOLLOWS.  IF TEST FAILS, SLENDER-BODY THEORY       AROX  1260
C     WILL BE USED AUTOMATICALLY.                                          AROX  1270
C                                                                          AROX  1280
      IF (ITYPE .GT. 1 .AND. (BETA * AR * (1.0 + LAMBDA) * (1.0 /         AROX  1290
     A    (BETA * M) + 1.0)) .LT. 4.0) ITYPE = 1                           AROX  1300
      I = ITYPE + 1                                                        AROX  1310
      GO TO (30, 2, 3, 5, 6, 8) , I                                        AROX  1320
C                                                                          AROX  1330
C     EQ. (21)                                                             AROX  1340
C                                                                          AROX  1350
    2 KBW = (    (1.0 - RS ** 2) ** 2 - 2.0 / PI * ( (1.0 + RS ** 4) *     AROX  1360
     A        (0.5 * ATAN (0.5 * (SR - RS)) + PI / 4.0) - RS ** 2 *        AROX  1370
     B        (SR - RS + 2.0 * ATAN (RS)) )    ) / (1.0 - RS) ** 2         AROX  1380
      GO TO 30                                                             AROX  1390
C                                                                          AROX  1400
C     EQ. (24)                                                             AROX  1410
C                                                                          AROX  1420
    3 BM1 = BETA * M                                                       AROX  1430
```

PLUNGE

PLUNGE

```
DECK AROX
      IF (BM1 .LT. 1.0) GO TO 4
      BM2 = BM1 ** 2
      BMR2 = SQRT (BM2 - 1.0)
      T1 = (    1.0 + (1.0 + BM1) * BETA * D / CR    )
     A     / (    BM1 + (BM1 + 1.0) * BETA * D / CR    )
      T2 = 1.0 / BM1
      T1 = ARCOS (T1)
      T2 = ARCOS (T2)
      KBW = 8.0 * BM1 / (PI * BMR2 * (1.0 + LAMBDA) * BETA * D / CR *
     A        (SR - 1.0) * BETA * CLALW)
     B        * (    (BM1 / (1.0 + BM1)) * (((BM1 + 1.0) * BETA * D / CR
     C        + BM1) / BM1) ** 2 * T1
     D        + BMR2 / (BM1 + 1.0) * (SQRT(1.0 + 2.0 * BETA * D / CR)
     E        - 1.0)
     F        - BMR2 / BM1 * (BETA * D / CR) ** 2 * ACOSH(1.0 + CR/BETA/D)
     G        - BM1 / (1.0 + BM1) * T2    )
      GO TO 30
C
C     EQ. (26)
C
    4 BM = BM1
      T1 = SQRT ( (BM + (1.0 + BM) * BETA * D / CR) / BM )
      KBW = 16.0 * (BM / (1.0 + BM)) ** 2
     A        / (PI * (1.0 + LAMBDA) * BETA * D / CR * (SR - 1.0) * BETA
     B        * CLALW)
     C        * (    T1 ** 3 + T1 - 2.0 - (((1.0 + BM) * BETA * C / CR)
     D        / BM) ** 2 * ATANH (1.0 / T1)    )
      GO TO 30
C
C     EQ. (27)
C
    5 BA = BETA * AR
      BARS = BA * RS
      T1 = BA / (BA + SR - 1.0)
      T1 = ARCOS (T1)
      KBW = 2.0 / (PI * (BA - 0.5)) *
```

```
DECK AROX

      A      (    0.5 * (1.0 + BARS / (1.0 - RS)) ** 2 * T1                     AROX
      B      - 0.5 * (BARS / (1.0 - RS)) ** 2 * ACOSH (1.0 + (1.0 - RS)         AROX
      C      / BARS) - 0.5 - PI / 4.0 + 0.5 * SQRT (1.0 + 2.0 * BARS            AROX
      D      / (1.0 - RS))    )                                                 AROX
       GO TO 30                                                                 AROX
C                                                                               AROX
C      EQ. (28)                                                                 AROX
C                                                                               AROX
     6 BA = BETA * AR                                                           AROX
       BA4 = BA / 4.0                                                           AROX
       IF (BA4 .GE. 1.0) GO TO 7                                                AROX
       T1 = SQRT (1.0 - BA4 ** 2)                                               AROX
       CALL ELPI (T1, DUMMY, T1, NER1)                                          AROX
C      ERROR TEST 3                                                             AROX
       IF (NER1.NE.0) WRITE (6,22)                                              AROX
   22  FORMAT (1H ,46H**** ELLIPTICAL INTEGRAL ERROR. T1 FROM PLUNGE            AROX
     1  55H IS NOT LESS THAN ONE AND GREATER THAN OR EQUAL TO ZERO )            AROX
       T2 = (BA4 / (BA4 + 1.0)) ** 2 / 2.0                                      AROX
       T3 = SQRT (1.0 + 2.0 * (1.0 + BA4) * RS / (1.0 - RS))                    AROX
       KBW = 8.0 * T1 / (PI * BA4) ** 2 *                                       AROX
      A     (    T2 * T3 ** 3 - (BA / (BA + 4.0)) ** 2 + T2 * T3                AROX
      B      - 2.0 * (BA4 * RS / (1.0 - RS)) ** 2 * ATANH (1.0 / T3)    )       AROX
       GO TO 30                                                                 AROX
C                                                                               AROX
C      EQ. (29)                                                                 AROX
C                                                                               AROX
     7 T1 = 2.0 * (1.0 + BA4) * RS / (1.0 - RS)                                 AROX
       T2 = (1.0 + BA4 * T1) / (BA4 + BA4 * T1)                                 AROX
       T3 = 1.0 / BA4                                                           AROX
       T2 = ARCOS (T2)                                                          AROX
       T3 = ARCOS (T3)                                                          AROX
       KBW = 1.0 / (PI * SQRT (BA4 ** 2 - 1.0)) *                               AROX
      A      (    (BA / (BA + 4.0)) * (1.0 + T1) ** 2 * T2                      AROX
      B      + SQRT ( (BA4 ** 2 - 1.0) * (1.0 + BA * RS / (1.0 - RS)) )         AROX
      C      / (1.0 + BA4)  - BA4 * T3 / (1.0 + BA4)                            AROX
      D      - SQRT (BA4 ** 2 - 1.0) * BA * (RS / (1.0 - RS)) ** 2              AROX
```

PLUNGE

DECK AROX

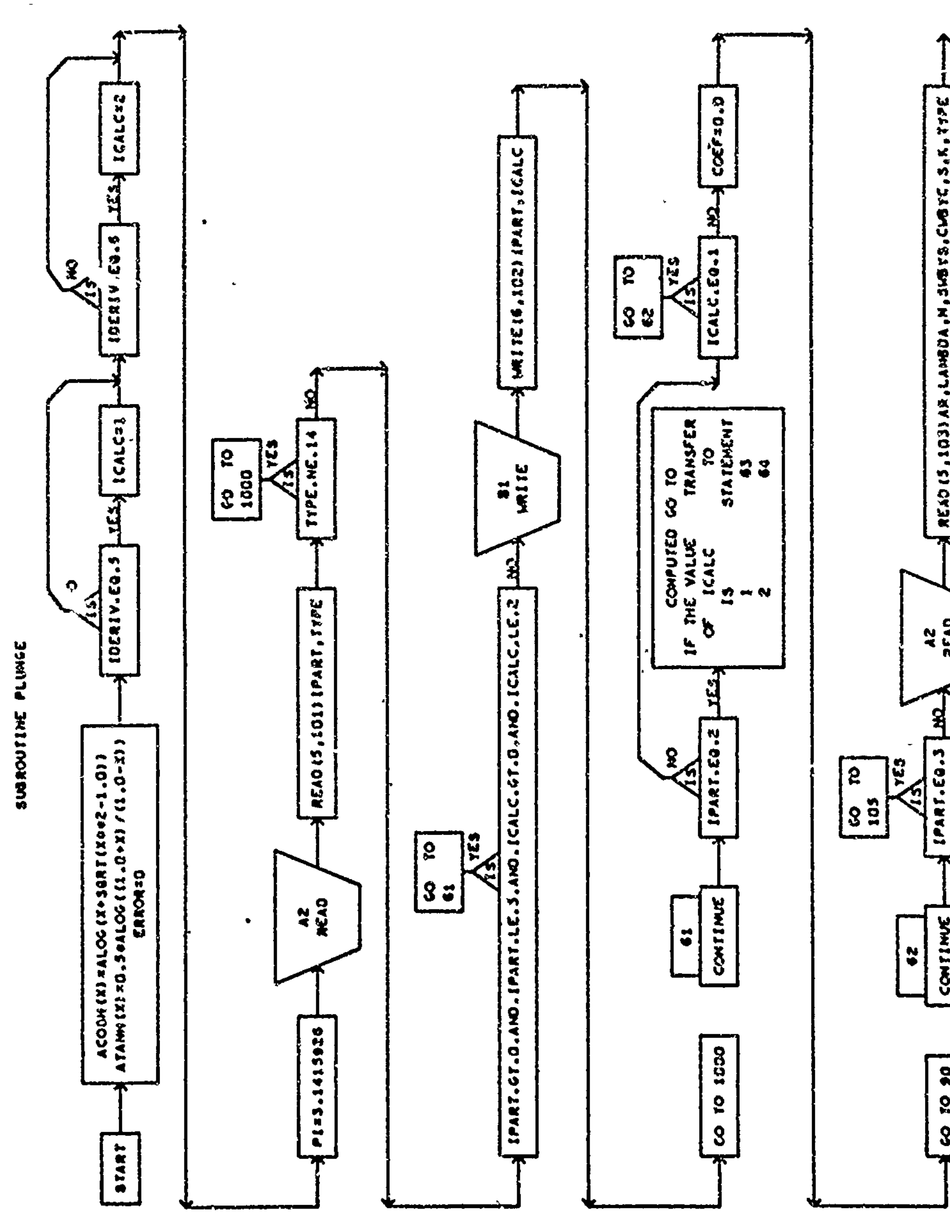
```
     E      * ACOSH (1.0 + 2.0 * (1.0 - RS) / BA / RS)
     F      - SQRT (BA4 ** 2 - 1.0) / (BA4 + 1.0)    )
      GO TO 30
C
C     EQNS. (30) AND (31)
C
    8 BDCR = BETA * D / CR
C
C     TEST BDCR.  IF BDCR GT 1.0, SET BDCR = 1.0 TO GET PROPER RESULT.
C     SEE PARAGRAPH FOLLOWING EQ. (31) OF NACA REPORT 1307.
C
      IF (BDCR .GT. 1.0) BDCR = 1.0
      BM    = BETA * M
      IF (BM .LT. 1.0) GO TO 9
C
C     EQ. (30)
C
      T1 = (BM + 1.0 / BDCR) / (1.0 + BM / BDCR)
      T2 = 1.0 / BM
      T3 = BDCR
      T4 = SQRT (BM ** 2 - 1.0)
      T1 = ARCOS (T1)
      T2 = ARCOS (T2)
      T3 = ARSIN (T3)
      KBW = 8.0 * BDCR / (PI * T4 * BETA * CLALW * (1.0 + LAMBDA) *
     A      (SR - 1.0)) *
     B      (   (1.0 + BM / BDCR ) ** 2 * T1
     C      - (BM / BDCR) ** 2 * T2
     D      + BM / BDCR ** 2 * T4 * T3
     E      - T4 * ACOSH (1.0 / BDCR)
      GO TO 30
    9 CRBD = 1.0 / BDCR
      CRBD2 = CRBD ** 2
C
C     FQ. (31)
C
```

```
DECK AROX

      KBW = 16.0 * SQRT (BM) * BDCR / (PI * (BM + 1.0) * BETA * CLALW      AROX  2520
     A      * (1.0 + LAMBDA) * (SR - 1.0)) *                               AROX  2530
     B      (    (1.0 + M * CR / D) * SQRT ((CRBD - 1.0)*(M*CR/D + 1.0))   AROX  2540
     C      - CRBD2 * BM ** 1.5                                            AROX  2550
     D      + BM * CRBD2 * (BM + 1.0) * (ATAN (SQRT(1.0 / BM))             AROX  2560
     E      - ATAN (SQRT((CRBD - 1.0) / (M * CR / D + 1.0))))              AROX  2570
     F      - (BM + 1.0) / SQRT (BM) * ATANH (SQRT(BM * (CRBD - 1.0) /     AROX  2580
     G        (M * CR / D + 1.0)))    )                                    AROX  2590
      GO TO 30                                                             AROX  2600
C                                                                          AROX  2610
C     HOPEFULLY, BY THIS TIME, KBW HAS BEEN CALCULATED.                    AROX  2620
C                                                                          AROX  2630
   30 CONTINUE                                                             AROX  2640
      IF (IPART .EQ. 3) GO TO 32                                           AROX  2650
C                                                                          AROX  2660
C***** CALCULATE WING CM ALPHA DOT ***********************************     AROX  2670
      COEF = SWBYS * (KWB + KBW) * CWBYC ** 2 * CMWPR                      AROX  2680
      GO TO 90                                                             AROX  2690
C                                                                          AROX  2700
C                                                                          AROX  2710
C***** CALCULATE TAIL CONTRIBUTION TO C-SUB-M-SUB-BETA-DOTA                AROX  2720
   32 CONTINUE                                                             AROX  2730
      COEF = 2.0 * Q * SWBYS * ( COS (0.017453292 * GAMMAT) ) ** 2         AROX  2740
     A       * CLALW * (KWB + KBW) * XBTBYC ** 2 * UPWASH                  AROX  2750
      GO TO 90                                                             AROX  2760
C                                                                          AROX  2770
C                                                                          AROX  2780
C***** CONTRIBUTION OF BODY TO C-SUB-M-SUB-ALPHA-DOT   *****************   AROX  2790
   63 CONTINUE                                                             AROX  2800
C                                                                          AROX  2810
C     READ BODY DATA                                                       AROX  2820
C                                                                          AROX  2830
      READ (5,104) VOLUME , SFRONT , LENGTH , XO , XC , C , TYPE           AROX  2840
      IF (TYPE .NE. 20) GO TO 1000                                         AROX  2850
      RATIO = (-2.0 * VOLUME / C / SFRONT) * (XO - XC) / LENGTH            AROX  2860
     A        / (VOLUME / SFRONT / LENGTH - 1.0 + XO / LENGTH)             AROX  2870
```

PLUNGE

DECK AROX

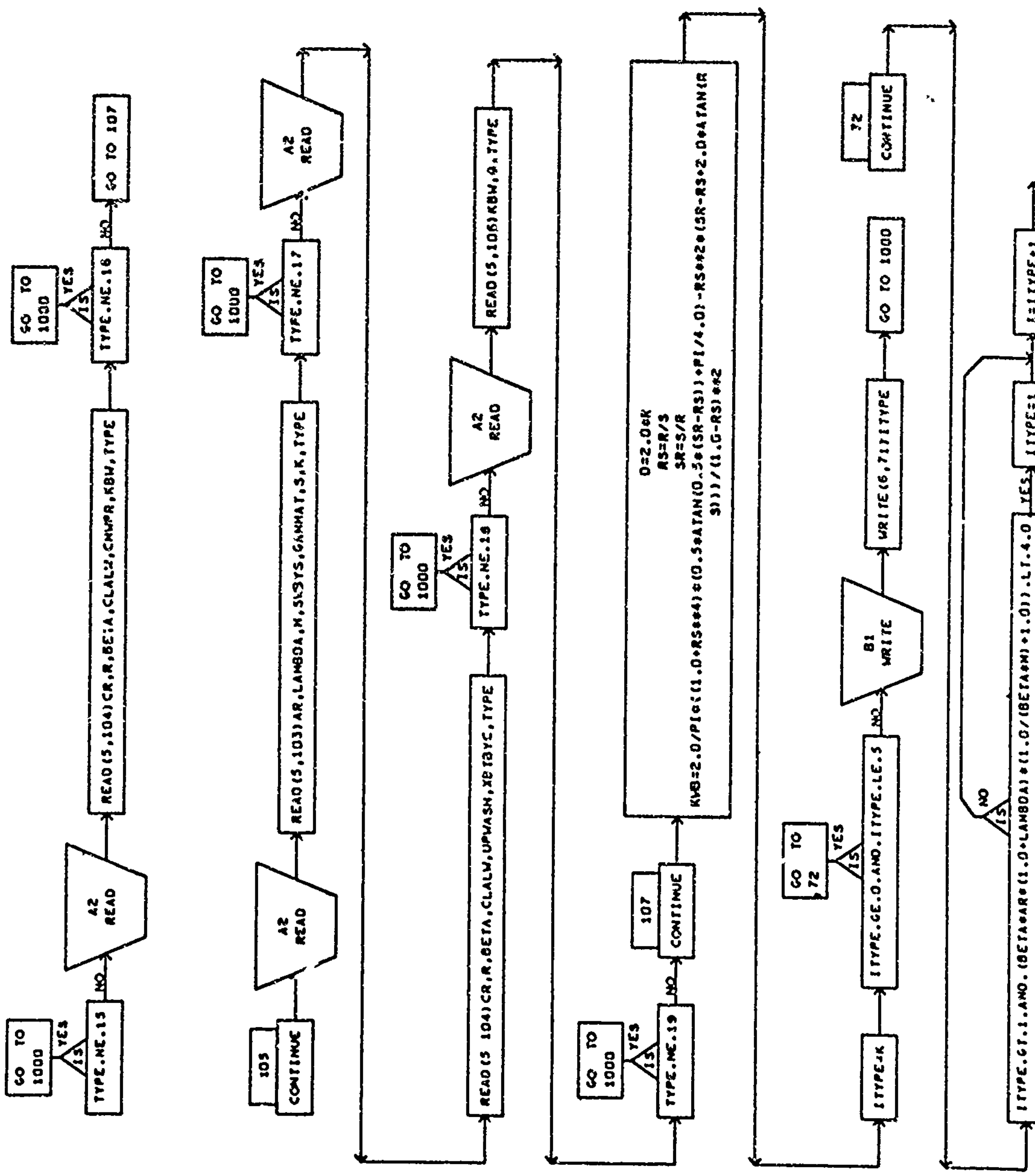
```
      COEF = RATIO * CMA                                               AROX
      GO TO 90                                                         AROX
C                                                                      AROX
C***** BODY CY BETA DOT *********************************************** AROX
   64 CONTINUE                                                         AROX
C     THIS PART OF THE ROUTINE COMPUTES C-SUB-Y-SUB-BETA DOT           AROX
C     THE BASIC REFERENCE IS NASA TMX-287.                             AROX
C     THE REFERENCE SHOWS THAT WING AND TAIL CONTRIBUTIONS.CAN BE      AROX
C     NEGLECTED, AND THAT THE FUSELAGE TERM CAN BE OBTAINED BY A       AROX
C     SLENDER-BODY-THEORY RATIO MULTIPLIED BY C-SUB-Y-SUB-BETA.        AROX
C                                                                      AROX
C                                                                      AROX
C     READ BODY DATA                                                   AROX
C                                                                      AROX
      READ (5,108) VOLUME , SFRONT , B , TYPE                          AROX
  108 FORMAT (3F10.0 , 40X, I2)                                        AROX
      IF (TYPE .NE. 21) GO TO 1000                                     AROX
      COEF = (-2.0) * VOLUME / SFRONT * CYB / B                        AROX
   90 IF (IDERIV .EQ. 5) CMADT = COEF                                  AROX
      IF (IDERIV .EQ. 6) CYBDT = COEF                                  AROX
      RETURN                                                           AROX
 1000 ERROR = 1                                                        AROX
      WRITE (6,1001)                                                   AROX
 1001 FORMAT (1H , 39H**** SUBROUTINE PLUNGE SETS ERROR FLAG ///)      AROX
      RETURN                                                           AROX
      END                                                              AROX
```



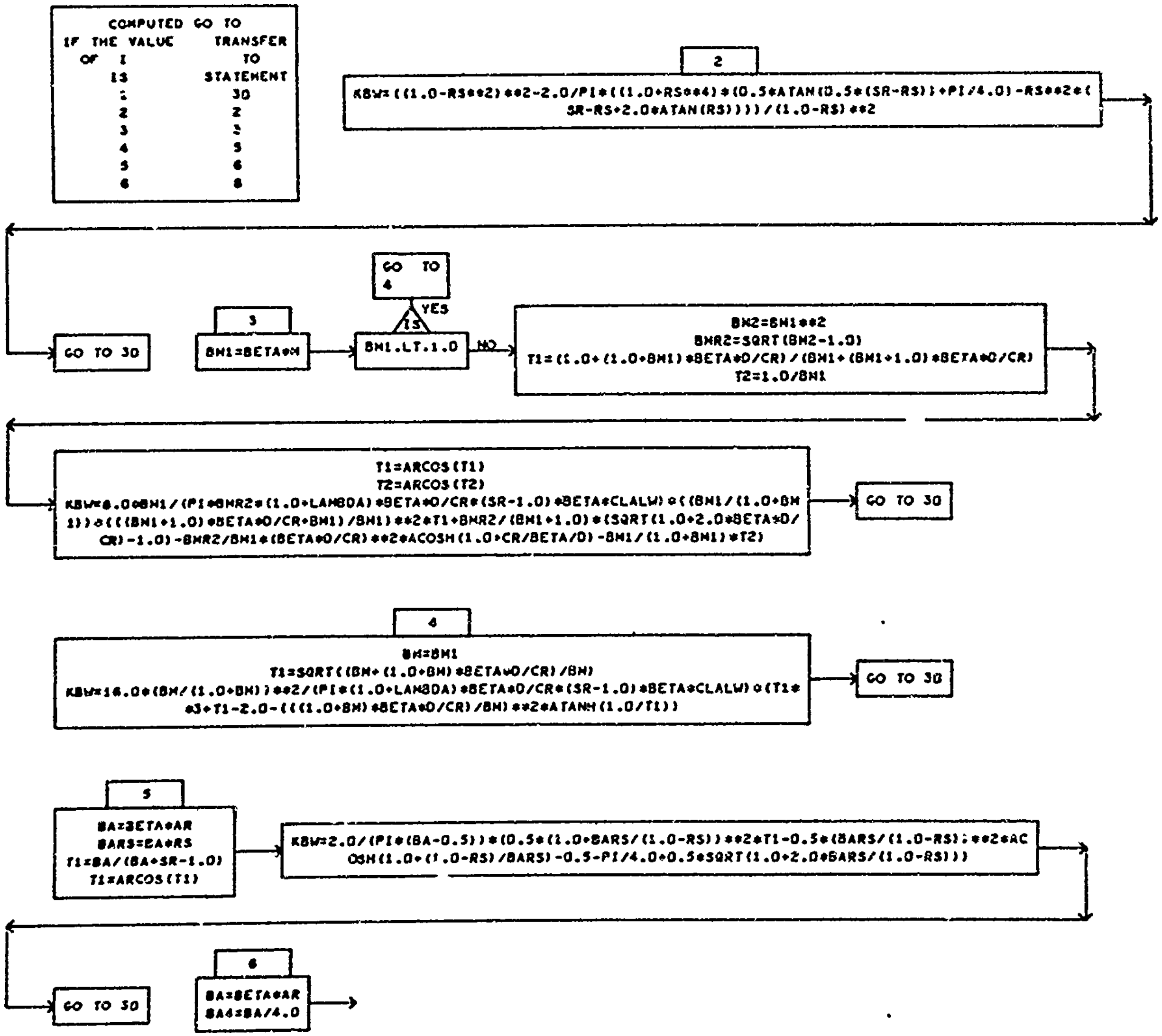
SUBROUTINE PLUNGE
START
ACOSH(X)=ALOG(X+SQRT(X**2-1.0))
ATANH(X)=0.5*ALOG((1.0+X)/(1.0-X))
ERROR=0
IS
IDERIV.EQ.5
NO
YES
ICALC=1
IS
IDERIV.EQ.6
NO
YES
ICALC=2
PI=3.1415926
A2
READ
READ(5,101)IPART,TYPE
IS
TYPE.NE.14
YES
GO TO 1000
NO
IPART.GT.0.AND.IPART.LE.5.AND.ICALC.GT.0.AND.ICALC.LE.2
IS
YES
GO TO 61
NO
B1
WRITE
WRITE(6,102)IPART,ICALC
GO TO 1000
61
CONTINUE
IS
IPART.EQ.2
NO
YES
COMPUTED GO TO
IF THE VALUE OF ICALC IS
1
2
TRANSFER TO STATEMENT
63
64
IS
ICALC.EQ.1
YES
GO TO 62
NO
COEF=0.0
GO TO 90
62
CONTINUE
IS
IPART.EQ.3
YES
GO TO 105
NO
A2
READ
READ(5,103)AR,LAMBDA,M,SWBYS,CWBYC,S,K,TYPE

PLUNGE

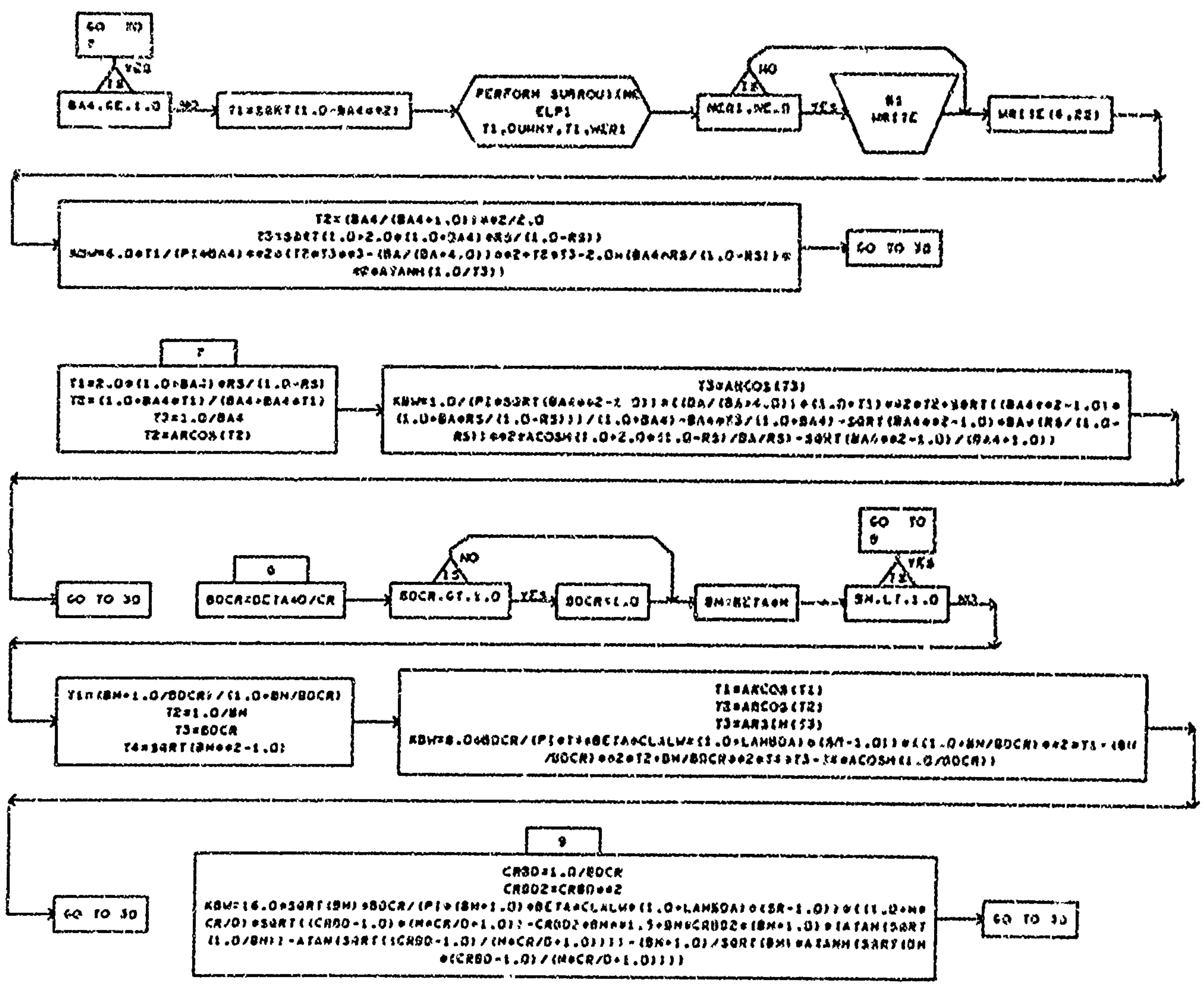
PLUNGE

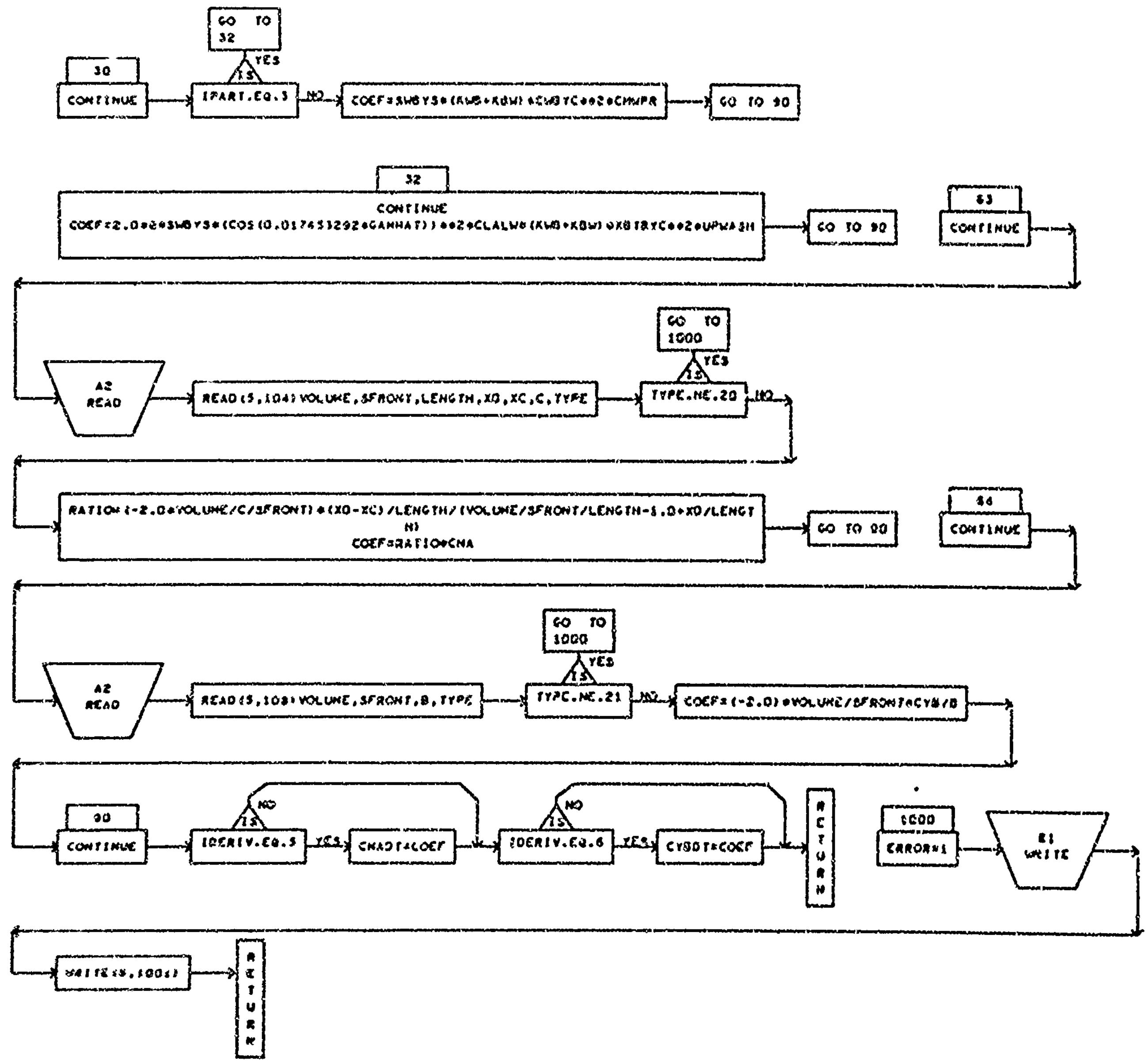


GO TO 1000
YES
IS
TYPE.NE.15
NO
A2 READ
READ(5,104)CR,R,BETA,CLALW,CMWPR,KBW,TYPE
GO TO 1000
YES
IS
TYPE.NE.16
NO
GO TO 107
105 CONTINUE
A2 READ
READ(5,103)AR,LAMBDA,M,SLSYS,GAMMAT,S,K,TYPE
GO TO 1000
YES
IS
TYPE.NE.17
NO
A2 READ
READ(5,104)CR,R,BETA,CLALW,UPWASH,XETSYS,TYPE
GO TO 1000
YES
IS
TYPE.NE.18
NO
A2 READ
READ(5,106)KBW,Q,TYPE
GO TO 1000
YES
IS
TYPE.NE.19
NO
107 CONTINUE
D=2.0*K
RS=R/S
SR=S/R
KWB=2.0/PI*((1.0+RS**4)*(0.5*ATAN(0.5*(SR-RS))+PI/4.0)-RS**2*(SR-RS+2.0*ATAN(RS)))/(1.0-RS)**2
ITYPE=K
GO TO 72
YES
IS
ITYPE.GE.0.AND.ITYPE.LE.5
NO
B1 WRITE
WRITE(6,71)ITYPE
GO TO 1000
72 CONTINUE
ITYPE.GT.1.AND.(BETA*AR*(1.0+LAMBDA)*(1.0/(BETA*M)+1.0)).LT.4.0
IS
NO
YES
ITYPE=1
J=ITYPE+1


COMPUTED GO TO
IF THE VALUE OF I IS
TRANSFER TO STATEMENT
1 30
2 2
3 3
4 5
5 6
6 8
2
KBW=((1.0-RS**2)**2-2.0/PI*((1.0+RS**4)*(0.5*ATAN(0.5*(SR-RS))+PI/4.0)-RS**2*(
SR-RS+2.0*ATAN(RS))))/(1.0-RS)**2
GO TO 30
3
BM1=BETA*M
BM1.LT.1.0
IS
YES
NO
GO TO 4
BM2=BM1**2
BMR2=SQRT(BM2-1.0)
T1=(1.0+(1.0+BM1)*BETA*D/CR)/(BM1+(BM1+1.0)*BETA*D/CR)
T2=1.0/BM1
T1=ARCOS(T1)
T2=ARCOS(T2)
KBW=8.0*BM1/(PI*BMR2*(1.0+LAMBDA)*BETA*D/CR*(SR-1.0)*BETA*CLALW)*((BM1/(1.0+BM
1))*(((BM1+1.0)*BETA*D/CR+BM1)/BM1)**2*T1+BMR2/(BM1+1.0)*(SQRT(1.0+2.0*BETA*D/
CR)-1.0)-BMR2/BM1*(BETA*D/CR)**2*ACOSH(1.0+CR/BETA/D)-BM1/(1.0+BM1)*T2)
GO TO 30
4
BM=BM1
T1=SQRT((BM+(1.0+BM)*BETA*D/CR)/BM)
KBW=16.0*(BM/(1.0+BM))**2/(PI*(1.0+LAMBDA)*BETA*D/CR*(SR-1.0)*BETA*CLALW)*(T1*
*3+T1-2.0-(((1.0+BM)*BETA*D/CR)/BM)**2*ATANH(1.0/T1))
GO TO 30
5
BA=BETA*AR
BARS=BA*RS
T1=BA/(BA+SR-1.0)
T1=ARCOS(T1)
KBW=2.0/(PI*(BA-0.5))*(0.5*(1.0+BARS/(1.0-RS))**2*T1-0.5*(BARS/(1.0-RS))**2*AC
OSH(1.0+(1.0-RS)/BARS)-0.5-PI/4.0+0.5*SQRT(1.0+2.0*BARS/(1.0-RS)))
GO TO 30
6
BA=BETA*AR
BA4=BA/4.0

PLUNGE


GO TO 7
YES
BA4.GE.1.0
NO
T1=SQRT(1.0-BA4**2)
PERFORM SUBROUTINE ELP1
T1,DUMMY,T1,NER1
NO
NER1.NE.0
YES
N1 WRITE
WRITE(6,22)
T2=(BA4/(BA4+1.0))**2/2.0
GO TO 30
7
T2=ARCOS(T2)
T3=ARCOS(T3)
GO TO 30
8
BOCR=BETA4D/CR
NO
BOCR.GT.1.0
YES
BOCR=1.0
BH=BETA4H
GO TO 9
YES
BH.LT.1.0
NO
GO TO 30
T2=1.0/BH
T3=BOCR
T4=SQRT(BH**2-1.0)
T1=ARCOS(T1)
T2=ARCOS(T2)
9
CRBD=1.0/BOCR
CRBD2=CRBD**2
GO TO 30
GO TO 30


30
CONTINUE
IPART.EQ.3
IS
YES
GO TO 32
NO
COEF=SWBYS*(KWB+KBW)*CWBYC**2*CMWPR
GO TO 90
32
CONTINUE
COEF=2.0*2*SWBYS*(COS(0.017453292*GAMMAT))**2*CLALW*(KWB+KBW)*XBTRYC**2*UPWASH
GO TO 90
63
CONTINUE
A2
READ
READ(5,104)VOLUME,SFRONT,LENGTH,XO,XC,C,TYPE
TYPE.NE.20
IS
YES
GO TO 1000
NO
RATIO=(-2.0*VOLUME/C/SFRONT)*(XO-XC)/LENGTH/(VOLUME/SFRONT/LENGTH-1.0+XO/LENGT
H)
COEF=RATIO*CNA
GO TO 90
64
CONTINUE
A2
READ
READ(5,108)VOLUME,SFRONT,B,TYPE
TYPE.NE.21
IS
YES
GO TO 1000
NO
COEF=(-2.0)*VOLUME/SFRONT*CYB/B
90
CONTINUE
IDERIV.EQ.5
IS
NO
YES
CNADT=COEF
IDERIV.EQ.6
IS
NO
YES
CYBDT=COEF
RETURN
1000
ERROR=1
E1
WRITE
WRITE(6,1001)
RETURN

PLUNGE

SYMBOLS USED IN SUBROUTINE PLUNGE

AR	R	U	ASPECT RATIO OF WING/TAIL	PLUNGE
B	R	U	WING/TAIL SPAN	PLUNGE
BA	R	U	PRODUCT OF BETA AND ASPECT RATIO	PLUNGE
BARS	R	U	PRODUCT OF BA AND RS	PLUNGE
BA4	R	U	BA DIVIDED BY 4	PLUNGE
BDCR	R	U	PRODUCT OF BETA AND D DIVIDED BY CR	PLUNGE
BETA	R	U	PRANDTL-GLAUERT FACTOR	PLUNGE
BM	R	U	PRODUCT OF BETA AND M	PLUNGE
BMR2	R	U	SQUARE ROOT OF DIFFERENCE OF BM2 AND 1.0	PLUNGE
BM1	R	U	PRODUCT OF BETA AND M	PLUNGE
BM2	R	U	SQUARE OF PRODUCT OF BETA AND M	PLUNGE
C	R	U	REFERENCE CHORD FOR BODY	PLUNGE
CASE	I	C	CASE NUMBER	PLUNGE
CLALW	R	U	LIFT-CURVE SLOPE FOR WING/TAIL (PER. RADIAN)	PLUNGE
CMA	R	A	DERIVATIVE OF PITCHING MOMENT WITH ANGLE OF ATTACK	PLUNGE
CMADT	R	A	PITCHING MOMENT-ALPHA DOT DERIVATIVE	PLUNGE
CMWPR	R	U	WING-ALONE/TAIL-ALONE PITCHING MOMENT DERIVATIVE	PLUNGE
COEF	R	U	COEFFICIENT DERIVATIVE	PLUNGE
CR	R	U	WING/TAIL CHORD AT WING/TAIL-BODY JUNCTURE	PLUNGE
CRBD	R	U	RICIPROCAL OF BDCR	PLUNGE
CRBD2	R	U	SQUARE OF CRBD	PLUNGE
CWBYC	R	U	MEAN AERODYNAMIC CHORD OF EXPOSED WING/TAIL	PLUNGE
CYB	R	A	DERIVATIVE OF SIDE FORCE WITH YAW ANGLE	PLUNGE
CYBDT	R	A	DERIVATIVE OF SIDE FORCE WITH BETA DOT	PLUNGE
D	R	U	BODY DIAMETER AT WING OR TAIL	PLUNGE
DUMMY	R	U	DUMMY VARIABLE	PLUNGE
ERROR	I	C	ERROR FLAG	PLUNGE
GAMMAT	R	U	TAIL DIHEDRAL ANGLE (DEGREES)	PLUNGE
I	I	U	INDEX	PLUNGE
ICALC	I	U	CALCULATION OPTION FLAG	PLUNGE
IDERIV	I	A	DERIVATIVE OPTION FLAG	PLUNGE
IPART	I	U	CONTROL FLAG FOR COMPONENT TYPE (BODY,WING,TAIL)	PLUNGE
ITYPE	I	U	FLAG TO CONTROL EQUATION TO BE USED IN CALCULATING KBW	PLUNGE
K	I	U	FLAG TO CONTROL SELECTION OF KBW EQUATION	PLUNGE
KBW	R	U	INTERFERENCE ON BODY IN PRESENCE OF WING/TAIL	PLUNGE
KWB	R	U	INTERFERENCE ON WING/TAIL IN PRESENCE OF BODY	PLUNGE

SYMBOLS USED IN SUBROUTINE PLUNGE

LAMBDA	R	U	WING/TAIL TAPER RATIO (TIP CHORD/ROOT CHORD)	PLUNGE
LENGTH	R	U	BODY LENGTH	PLUNGE
M	R	U	COTANGENT OF WING/TAIL LEADING EDGE SWEEP ANGLE	PLUNGE
NER1	I	U	ERROR FLAG	PLUNGE
PAGE	I	C	PAGE NUMBER	PLUNGE
PI	R	U	RATIO OF CIRCUMFERENCE OF A CIRCLE TO ITS DIAMETER	PLUNGE
Q	R	U	TAIL EFFECTIVENESS RATIO	PLUNGE
R	R	U	BODY RADIUS AT WING OR TAIL	PLUNGE
RATIO	R	U	DUMMY VARIABLE	PLUNGE
RS	R	U	R DIVIDED BY S	PLUNGE
S	R	U	WING/TAIL SEMI-SPAN	PLUNGE
SFRONT	R	U	BODY FRONTAL AREA	PLUNGE
SR	R	U	S DIVIDED BY R	PLUNGE
SWBYS	R	U	WING/TAIL AREA DIVIDED BY REFERENCE AREA	PLUNGE
TITLE	R	C	TITLE	PLUNGE
TYPE	I	U	CARD TYPE	PLUNGE
T1	R	U	DUMMY VARIABLE	PLUNGE
T2	R	U	DUMMY VARIABLE	PLUNGE
T3	R	U	DUMMY VARIABLE	PLUNGE
T4	R	U	DUMMY VARIABLE	PLUNGE
UPWASH	R	U	TAIL UPWASH DERIVATIVE CAUSED BY WING	PLUNGE
VOLUME	R	U	BODY VOLUME	PLUNGE
XBTBYC	R	U	TAIL LENGTH DIVIDED BY REFERENCE CHORD	PLUNGE
XC	R	U	AREA CENTROID LOCATION OF BODY	PLUNGE
XO	R	U	CENTER OF GRAVITY LOCATION	PLUNGE

26. SUBROUTINE ELP1 (DECK AROY)

This routine approximates the values of the elliptical integrals of the first and second kinds.

a. Algorithm

The approximation to a value of the elliptical integral of the first kind is given by

$$K(k) = (a_o + a_1\eta + \ldots + a_4\eta^4) + (b_o + b_1\eta + \ldots + b_4\eta^4) \ln \frac{1}{\eta}$$

where $\eta = 1 - k^2$

$a_o = 1.386294361$ $\quad$ $b_o = 0.5$

$a_1 = 0.0966634426$ $\quad$ $b_1 = 0.124985936$

$a_2 = 0.0359009238$ $\quad$ $b_2 = 0.0688024857$

$a_3 = 0.0374256371$ $\quad$ $b_3 = 0.0332835534$

$a_4 = 0.0145119621$ $\quad$ $b_4 = 0.0044178701$

The approximation to the value of the elliptical integral of the second kind is given by

$$E(k) = (a_o + a_1\eta + \ldots + a_4\eta^4) + (b_o + b_1\eta + \ldots + b_4\eta^4) \ln \frac{1}{\eta}$$

where $\eta = 1 - k^2$

$a_o = 1.0$ $\quad$ $b_o = 0.0$

$a_1 = 0.4432514146$ $\quad$ $b_1 = 0.2499836831$

$a_2 = 0.0626060122$ $\quad$ $b_2 = 0.0920018004$

$a_3 = 0.0475738355$ $\quad$ $b_3 = 0.0406969753$

$a_4 = 0.0173650645$ $\quad$ $b_4 = 0.0052644964$

b. Input/Output

None

c. Error

None

d. Subroutines Required

None

e. Argument List

(AK, AKK, E, NERROR)

f. Length

828 bytes

[redacted]

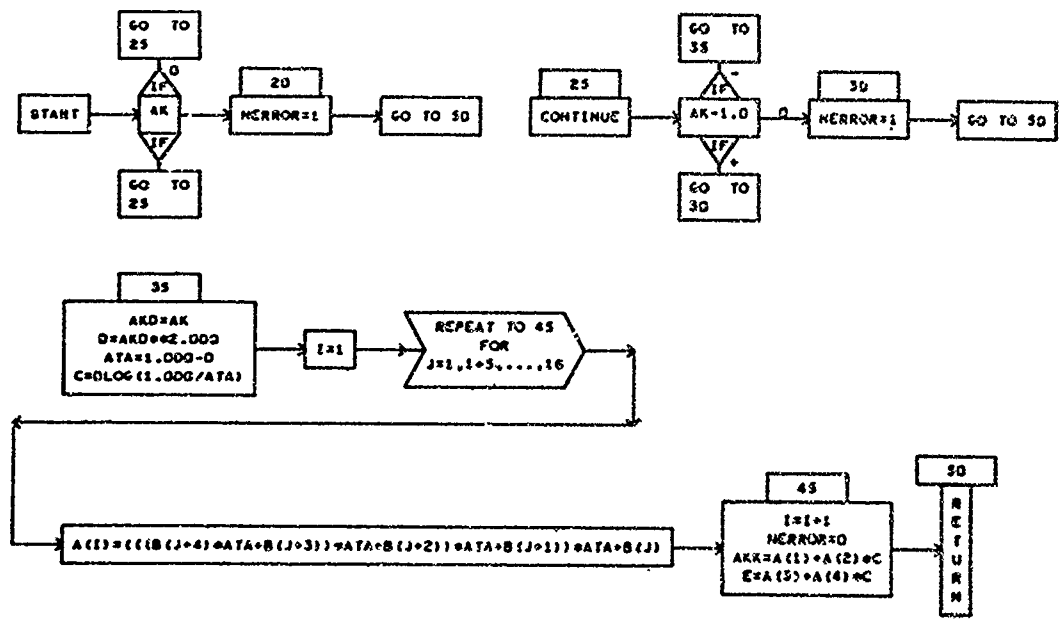
DECK AROY

```
      SUBROUTINE ELP1(AK, AKK, E, NERROR)                                      AROY  0010
CELP1  ELP1 IS A ROUTINE TO APPROXIMATE THE VALUE OF THE ELLIPTICAL            AROY  0020
C      INTEGRALS E(K) AND K(K) BY A METHOD OF NUMERICAL ANALYSIS.              AROY  0030
C      ARGUMENTS OF THE ROUTINE ARE AS FOLLOWS                                 AROY  0040
C      AK     = INPUT VALUE OF K                                               AROY  0050
C      AKK    = OUTPUT VALUE OF K(K) ELLIPTICAL INTEGRAL                       AROY  0060
C      E      = OUTPUT VALUE OF E(K) ELLIPTICAL INTEGRAL                       AROY  0070
C      NERROR = ERROR CODE                                                     AROY  0080
C               0 = K IS IN ALLOWABLE RANGE                                    AROY  0090
C               1 = K OUT OF ALLOWABLE RANGE, I.E. EQUAL TO OR GREATER         AROY  0100
C                   THAN ONE OR LESS THAN ZERO.                                AROY  0110
C                                                                              AROY  0120
      DIMENSION B(20), A(4)                                                    AROY  0130
      DOUBLE PRECISION A, B, ATA, C, D, AKD                                    AROY  0140
      DATA              B/1.38629436112, 0.09666344259, 0.03590092383,         AROY  0150
     1                                                                         AROY  0160
     2 .03742563713, .01451196212, .5D0, .12498593597, .06880248576,           AROY  0170
     3                                                                         AROY  0180
     4 .03328355346, .00441787C12D0, 1.0D0, .44325141463, .06260601220,        AROY  0190
     5                                                                         AROY  0200
     6 .C4757383546, .01736506451, .0D0, .29499836831, .09200180037,           AROY  0210
     7                                                                         AROY  0220
     8 .04C69657526, .00526449639D0/                                           AROY  0230
C                                                                              AROY  0240
      IF(AK) 20,25,25                                                          AROY  0250
   20 NERROR = 1                                                               AROY  0260
      GO TO 50                                                                 AROY  0270
   25 IF(AK-1.0) 35,30,30                                                      AROY  0280
   30 NERROR = 1                                                               AROY  0290
      GO TO 50                                                                 AROY  0300
   35 AKD = AK                                                                 AROY  0310
      D = AKD ** 2.0D0                                                         AROY  0320
      ATA = 1.0D0 - D                                                          AROY  0330
      C = DLOG(1.0D0 / ATA)                                                    AROY  0340
      I = 1                                                                    AROY  0350
```

DECK ARDY

```
      DO 45 J = 1,16,5                                                   ARDY  0360
      A(I) = (((B(J+4)*ATA + B(J+3))*ATA + B(J+2))*ATA + B(J+1))*ATA +   ARDY  0370
     1        B(J)                                                       ARDY  0380
   45 I = I + 1                                                          ARDY  0390
      NERROR = 0                                                         ARDY  0400
      AKK = A(1) + A(2)*C                                                ARDY  0410
      E   = A(3) + A(4)*C                                                ARDY  0420
   50 RETURN                                                             ARDY  0430
      END                                                                ARDY  0440
```

SUBROUTINE ELP1



SYMBOLS USED IN SUBROUTINE ELP1

A	D	D	ELLIPTICAL INTEGRAL PARAMETER	ELP1
AK	R	A	THE MODULUS	ELP1
AKD	D	U	MODULUS	ELP1
AKK	R	A	ELLIPTICAL INTEGRAL OF THE FIRST KIND	ELP1
ATA	D	U	ELLIPTICAL INTEGRAL PARAMETER	ELP1
B	D	D	ELLIPTICAL INTEGRAL CONSTANT ARRAY	ELP1
C	D	U	ELLIPTICAL INTEGRAL PARAMETER	ELP1
D	D	U	ELLIPTICAL INTEGRAL PARAMETER	ELP1
E	R	A	ELLIPTICAL INTEGRAL OF THE SECOND KIND	ELP1
I	I	U	INDEX	ELP1
J	I	U	DO-LOOP INDEX	ELP1
NERROR	I	A	ERROR FLAG	ELP1

27. SUBROUTINE VECTOR (DECK AROZ)

This routine converts input thrust vector data to coefficients and adds the results to the vehicle aerodynamic coefficients.

a. Algorithm

Set up initial conditions and constants. Read in a force vector and convert to force coefficients. Print the results if required. Continue to read in force vector data until LAST equals one.

b. Input/Output

Thrust Vector Data Cards (Type 22)
Thrust Vector coefficient contributions are printed if IPRINT = 1.

c. Error

An error condition occurs if the card type number is wrong.

d. Subroutines Required

HEADER

e. Argument List

(MACH, PFS, SREF, XCG, YCG, ZCG, SPAN, MAC, ALPHA, CD, CL, CA, CY, CN, BETA, LOD, CLM, CLL, CLN)

f. Length

2304 bytes

DECK AROZ

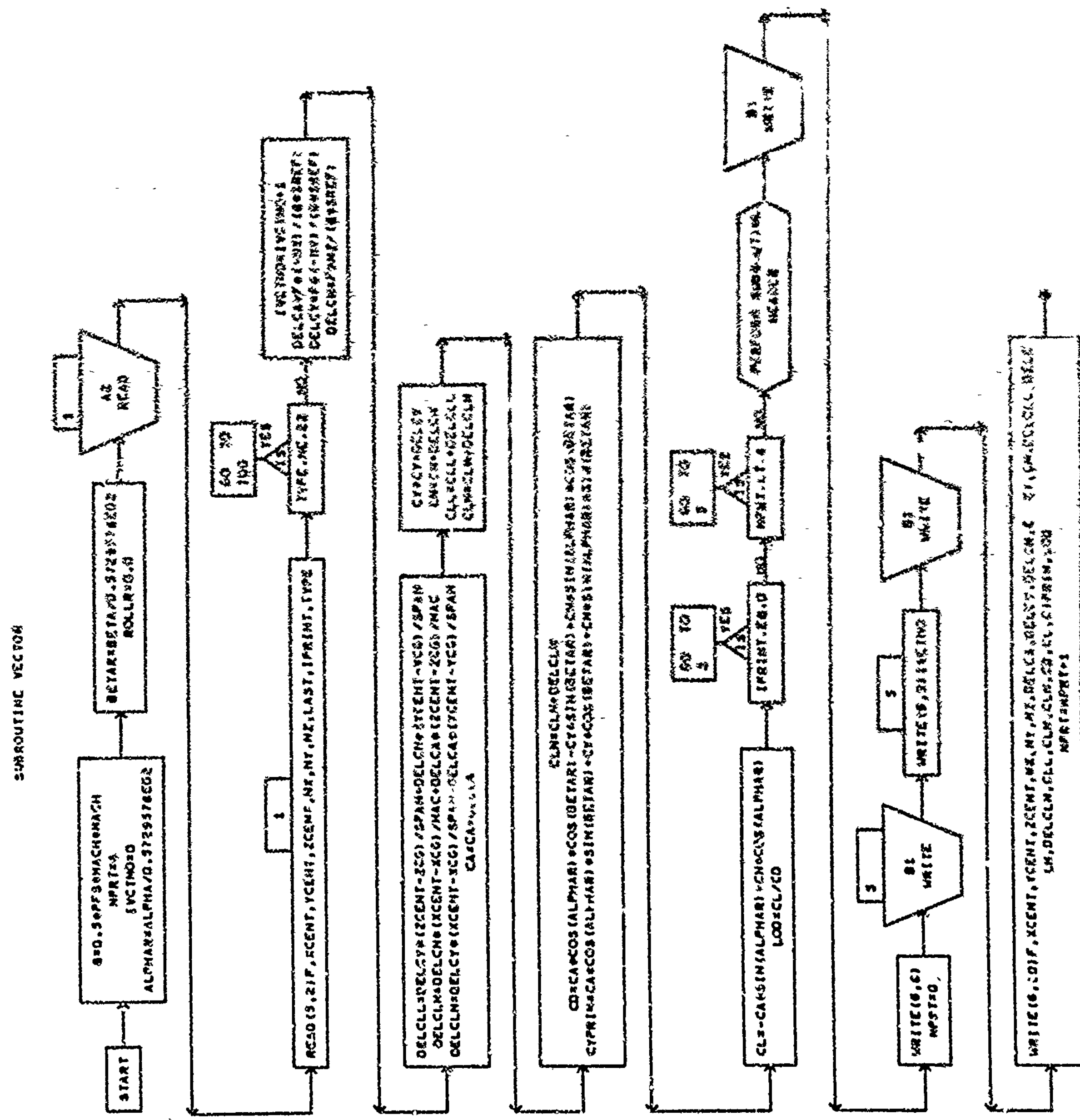
```
      SUBROUTINE VECTOR (MACH,PFS,SREF,XCG,YCG,ZCG,SPAN,MAC,
     1 ALPHA,CD,CL,CA,CY,CN,BETA,LOD,CLM,CLL,CLN)
C
C***************************************************************************
C**** THIS SUBROUTINE CONVERTS THRUST VECTORS INTO AERODYNAMIC
C**** COEFFICIENTS.  ANY NUMBER OF VECTORS MAY BE INPUT.
C***************************************************************************
C
      DIMENSION  TITLE(15)
      COMMON  CASE,TITLE,PAGE,ERROR
      REAL  MACH,MAC,LOD,NX,NY,NZ
      INTEGER  ERROR,CASE,PAGE,TYPE
C
C  SET UP INITIAL CONDITIONS
      Q = 0.5 * PFS * MACH*MACH
      NPRT = 4
      IVCTNO = 0
      ALPHAR = ALPHA / 0.5729578E02
      BETAR  = BETA  / 0.5729578E02
      ROLLR = 0.0
C
C  READ IN FORCE VECTOR DATA AND CONVERT TO FORCE COEFFICIENTS
    1 READ (5,2) F,XCENT,YCENT,ZCENT,NX,NY,NZ,LAST,IPRINT,TYPE
    2 FORMAT (F10.0,6F6.0,13X,I1,1XI1,8X,I2)
      IF (TYPE .NE. 22) GO TO 100
      IVCTNO = IVCTNO + 1
      DELCA = F *(-NX)/ (Q * SREF)
      DELCY = F *(-NY)/ (C * SREF)
      DELCN = F *  NZ / (Q * SREF)
      DELCLL= DELCY * (ZCENT - ZCG)/SPAN
     1       +DELCN * (YCENT - YCG)/SPAN
      DELCLM= DELCN * (XCENT - XCG)/MAC
     1       +DELCA * (ZCENT - ZCG)/MAC
      DELCLN= DELCY * (XCENT - XCG)/SPAN
     1       -DELCA * (YCENT - YCG)/SPAN
```

DECK AROZ

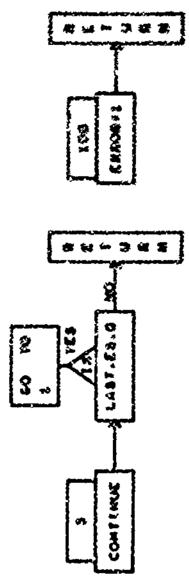
```
      CA   = CA   + DELCA                                               AROZ  0360
      CY   = CY   + DELCY                                               AROZ  0370
      CN   = CN   + DELCN                                               AROZ  0380
      CLL  = CLL  + DELCLL                                              AROZ  0390
      CLM  = CLM  + DELCLM                                              AROZ  0400
      CLN  = CLN  + DELCLN                                              AROZ  0410
      CD = CA*COS(ALPHAR)*COS(BETAR)  - CY*SIN(BETAR)                   AROZ  0420
     1    +CN*SIN(ALPHAR)*COS(BETAR)                                    AROZ  0430
      CYPRIM = CA*COS(ALPHAR)*SIN(BETAR)  + CY*COS(BETAR)               AROZ  0440
     1        +CN*SIN(ALPHAR)*SIN(BETAR)                                AROZ  0450
C                                                                       AROZ  0460
C                                                                       AROZ  0470
      CL  = -CA*SIN(ALPHAR) + CN*COS(ALPHAR)                            AROZ  0480
C                                                                       AROZ  0490
C                                                                       AROZ  0500
C                                                                       AROZ  0510
      LOD = CL / CD                                                     AROZ  0520
C                                                                       AROZ  0530
C    PRINT RESULTS IF REQUIRED                                          AROZ  0540
      IF (IPRINT .EQ. 0) GO TO 3                                        AROZ  0550
      IF (NPRT .LT. 4) GO TO 5                                          AROZ  0560
      CALL HEADER                                                       AROZ  0570
      WRITE (6,8)                                                       AROZ  0580
    8 FORMAT (1HO,50HRESULTS OF VECTOR CONVERSION TO FORCE COEFFICIENTS) AROZ  0590
      NPRT = 0                                                          AROZ  0600
    5 WRITE (6,9) IVCTNO                                                AROZ  0610
    9 FORMAT (1HO,13HVECTOR NUMBER,I3)                                  AROZ  0620
      WRITE (6,10) F,XCENT,YCENT,ZCENT,NX,NY,NZ,DELCA,DELCY,DELCN,      AROZ  0630
     1  CA,CY,CN,DELCLL,DELCLM,DELCLN,CLL,CLM,CLN,CD,CL,CYPRIM,LOD      AROZ  0640
   10 FORMAT (1H ,3X,3HF  =F12.1,2X6HXCENT=F7.1,2X6HYCENT=F7.1,         AROZ  0650
     1  2X6HZCENT=F7.1,/1H ,23X3HNX=F7.4,5X3HNX=F7.4,5X3HNX=F7.4,/      AROZ  0660
     2  1H ,3X8HDEL CA =E12.5,2X8HDEL CY =E12.5,2X8HDEL CN =E12.5,/     AROZ  0670
     3  1H ,3X8HTOT CA =E12.5,2X8HTOT CY =E12.5,2X8HTOT CN =E12.5,/     AROZ  0680
     4  1HO,3X8HDEL CLL=E12.5,2X8HDEL CLM=E12.5,2X8HDEL CLN=E12.5,/     AROZ  0690
     5  1H ,3X8HTOT CLL=E12.5,2X8HTOT CLM=E12.5,2X8HTOT CLN=E12.5,/     AROZ  0700
     6  1HO,6X5HC D =F9.5,8X5HC L =F9.5,7X,5HC Y =F9.5,/                AROZ  0710
```

DECK AROZ

```
    7   1H ,6X,5HL/D =F9.5)                                AROZ  0720
        NPRT = NPRT + 1                                    AROZ  0730
    3   IF (LAST .EQ. 0) GO TO 1                           AROZ  0740
        RETURN                                             AROZ  0750
  100   ERROR = 1                                          AROZ  0760
        RETURN                                             AROZ  0770
        END                                                AROZ  0780
```



VECTOR



VECTOR

SYMBOLS USED IN SUBROUTINE VECTOR

ALPHA	R	A	ANGLE OF ATTACK, DEGREES	VECTOR
ALPHAR	R	U	ANGLE OF ATTACK, RADIANS	VECTOR
BETA	R	A	YAW ANGLE, DEGREES	VECTOR
BETAR	R	U	YAW ANGLE, RADIANS	VECTOR
CA	R	A	AXIAL FORCE COEFFICIENT	VECTOR
CASE	I	C	CASE NUMBER	VECTOR
CD	R	A	DRAG COEFFICIENT	VECTOR
CL	R	A	LIFT COEFFICIENT	VECTOR
CLL	R	A	ROLLING MOMENT COEFFICIENT	VECTOR
CLM	R	A	PITCHING MOMENT COEFFICIENT	VECTUR
CLN	R	A	YAWING MOMENT COEFFICIENT	VECTOR
CN	R	A	NORMAL FURCE COEFFICIENT	VECTOR
CY	R	A	SIDE FORCE COEFFICIENT	VECTOR
CYPRIM	R	U	SIDE FORCE COEFFICIENT (WIND AXIS)	VECTOR
DELCA	R	U	DRAG COEFFICIENT INCREMENT	VECTOR
DELCLL	R	U	ROLLING MOMENT COEFFICIENT INCREMENT	VECTOR
DELCLM	R	U	PITCHING MOMENT COEFFICIENT INCREMENT	VECTOR
DELCLN	R	U	YAWING MOMENT COEFFICIENT INCREMENT	VECTOR
DELCN	R	U	NORMAL FORCE COEFFICIENT INCREMENT	VECTOR
DELCY	R	U	SIDE FORCE COEFFICIENT INCREMENT	VECTOR
ERROR	I	C	ERROR FLAG	VECTOR
F	R	U	FORCE MAGNITUDE, POUNDS	VECTOR
IPRINT	I	U	PRINT FLAG	VECTOR
IVCTNO	I	U	VECTOR NUMBER	VECTOR
LAST	I	U	LAST VECTOR FLAG	VECTOR
LOD	R	A	LIFT-TO-DRAG RATIO	VECTOR
MAC	R	A	REFERENCE LENGTH FOR MOMENT COEFFICIENTS	VECTOR
MACH	R	A	MACH NUMBER	VECTOR
NPRT	I	U	PRINT LINE COUNTER	VECTOR
NX	R	U	FORCE VECTOR DIRECTION COSINE IN X-DIRECTION	VECTOR
NY	R	U	FORCE VECTOR DIRECTION COSINE IN Y-DIRECTION	VECTOR
NZ	R	U	FORCE VECTOR DIRECTION COSINE IN Z-DIRECTION	VECTOR
PAGE	I	C	PAGE NUMBER	VECTOR
PFS	R	A	FREE-STREAM PRESSURE, LBS/SQUARE FOOT	VECTOR
Q	R	U	DYNAMIC PRESSURE	VECTOR
ROLLR	R	U	ROLL ANGLE IN RADIANS	VECTOR

SYMBOLS USED IN SUBROUTINE VECTOR

SPAN	R	A	REFERENCE LENGTH FOR ROLLING, YAWING COEFFICIENTS	VECTOR
SREF	R	A	VEHICLE REFERENCE AREA (WING AREA)	VECTOR
TITLE	R	C	TITLE	VECTOR
TYPE	I	U	CARD TYPE NUMBER	VECTOR
XCENT	R	U	ACTION POINT FOR FORCE VECTOR-X	VECTOR
XCG	R	A	X-CENTER FOR MOMENT CALCULATIONS	VECTOR
YCENT	R	U	ACTION POINT FOR FORCE VECTOR-Y	VECTOR
YCG	R	A	Y-CENTER FOR MOMENT CALCULATIONS	VECTOR
ZCENT	R	U	ACTION POINT FOR FORCE VECTOR-Z	VECTOR
ZCG	R	A	Z-CENTER FOR MOMENT CALCULATIONS	VECTOR

28. SUBROUTINE GRAPIC (DECK GRPA)

This is the Executive routine for the graphics part of the program.

a. Algorithm

Print that GRAPHIC OPTION HAS CONTROL and select the proper graphic routine depending upon the value of IPROG.

b. Input/Output

None

c. Error

None

d. Subroutines Required

PICTUR, PLOT

e. Argument List

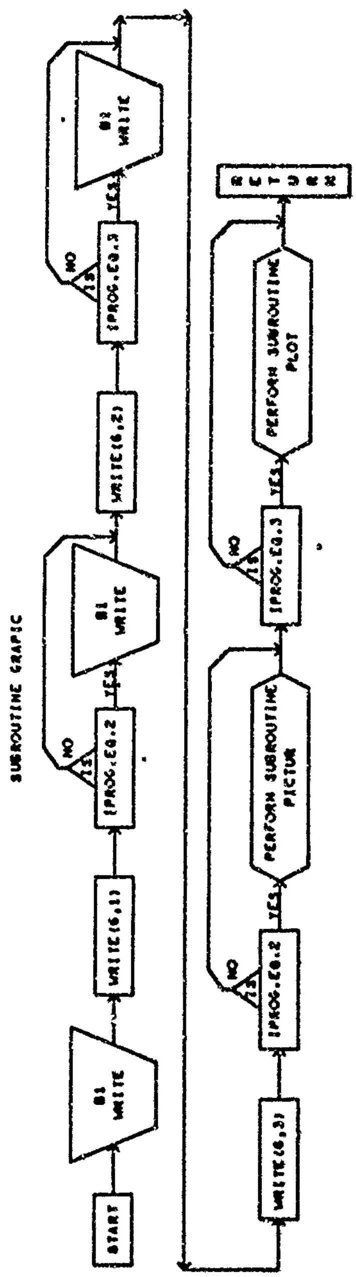
(IPROG)

f. Length

536 bytes

DECK GRPA

```
      SUBROUTINE GRAPIC (IPROG)
C
      COMMON CASE,TITLE,PAGE,ERROR
      DIMENSION  TITLE(15)
      INTEGER  ERROR,CASE,PAGE
C
C***** GRAPHIC OUTPUT DATA CONTROL PROGRAM ******************
C
C
      WRITE (6,1)
    1 FORMAT (1H ,///,1H ,38H***** GRAPHIC OPTION HAS CONTROL ***** )
      IF (IPROG .EQ. 2) WRITE (6,2)
    2 FORMAT (1H0,10X,40HPICTURE DRAWING PROGRAM WILL BE EXECUTED  )
      IF (IPROG .EQ. 3) WRITE (6,3)
    3 FORMAT (1H0,10X,44HOUTPUT DATA PLOTTER PROGRAM WILL BE EXECUTED )
C
      IF (IPROG.EQ. 2) CALL PICTUR
C
      IF (IPROG .EQ. 3) CALL PLOT
C
      RETURN
      END
```


SUBROUTINE GRAPIC
START
B1 WRITE
WRITE(6,1)
IPROG.EQ.2
IS
NO
YES
WRITE(6,2)
IPROG.EQ.3
WRITE(6,3)
PERFORM SUBROUTINE PICTUR
PERFORM SUBROUTINE PLOT
RETURN

SYMBOLS USED IN SUBROUTINE GRAPIC

CASE	I	C	CASE NUMBER	GRAPIC
ERROR	I	C	ERROR FLAG	GRAPIC
IPROG	I	A	PROGRAM OPTION NUMBER	GRAPIC
PAGE	I	C	PAGE NUMBER	GRAPIC
TITLE	R	C	TITLE	GRAPIC

29. SUBROUTINE PICTUR (DECK GRPB)

This routine prepares an output tape for procession on the SC-4020. The result will be pictures of the vehicle with the selected viewing angles.

a. Algorithm

Read the Picture Drawing Program Element Data Title Card (Type 31) and the Element Data Control Card (Type 32). Read the surface element data either from Tape 5 or from Tape 8 as directed. Read plotting instruction data (Card Types 34, 35, 36, and 37). Set up starting constants for pictures. Read element data from Tape 3 using the same techniques as for SDATA and convert to quadrilaterals. Generate scale grids if required. Plot points and draw lines between the points as directed by the input data. Print the detailed element characteristics if PRINTS is equal to 1.

b. Input/Output

Element Data Title Card (Type 31), Element Data Control Card (Type 32), Element Data Cards from Tape 5 or Tape 8 (Type 3), Picture Control Data Card (Type 34), Grid Data Card (Type 35), Scale Label Card (Type 36), and Plot Title Card(s) (Type 37). If PRINTS is equal to 1 the detailed element characteristics will be printed on Tape 6 (just as is the case for SDATA).

c. Error

An error condition occurs if a card type number is wrong.

d. Subroutines Required

HEADR2, SC-4020 routines

e. Argument List

None

f. Length

16180 bytes

DECK GRPB

```
      SUBROUTINE PICTUR                                                 GRPB  0010
C       SC-4020 PLOTTER PROGRAM FOR PLOTTING SURFACE DATA               GRPB  0020
C                                                                       GRPB  0030
C                                                                       GRPB  0040
      DIMENSION  XA(250),XB(250),YA(250),YB(250),ZA(250),ZB(250),       GRPB  0050
     1 XI(4),ETA(4),XIN(4),YIN(4),ZIN(4),VTITLE(8),HTITLE(8),HLABEL(15),GRPB  0060
     2 YIN2(4),ZIN2(4),CARD(20),TITLE(15)                               GRPB  0070
C                                                                       GRPB  0080
      COMMON  CASE,TITLE,PAGE,ERROR                                     GRPB  0090
C                                                                       GRPB  0100
      REAL  NX,NY,NZ,NXO                                                GRPB  0110
      LOGICAL  RFLAG, AFLAG, BFLAG                                      GRPB  0120
      INTEGER  STAT,STATT,TYPE,PRINTS,SYMFCT,CASE,PAGE,ERROR            GRPB  0130
C                                                                       GRPB  0140
C                                                                       GRPB  0150
      FIRST(QX,QY,QZ,Q1,Q2,Q3) = QX*Q1 + QY*Q2 + QZ*Q3                  GRPB  0160
C                                                                       GRPB  0170
      THIRD(QX,QY,QZ,QPSI,QTHETA,QPHI) = QX*(COS(QTHETA)*COS(QPSI)) +   GRPB  0180
     1  QY*(-SIN(QPSI)*COS(QPHI)+SIN(QTHETA)*COS(QPSI)*SIN(QPHI)) +     GRPB  0190
     2  QZ*(SIN(QPSI)*SIN(QPHI)+SIN(QTHETA)*COS(QPSI)*COS(QPHI))        GRPB  0200
C                                                                       GRPB  0210
      CALL CAMRAV (9)                                                   GRPB  0220
      IDUM = -1                                                         GRPB  0230
      REWIND 2                                                          GRPB  0240
      CALL INCRV (6,3)                                                  GRPB  0250
      IPIC = 1                                                          GRPB  0260
    2 REWIND 3                                                          GRPB  0270
C  READ ALL INPUT DATA AND STORE ON TAPE 3 FOR FUTURE USE               GRPB  0280
      READ(5,100) (TITLE(I),I=1,15),CASE,TYPE                           GRPB  0290
  100 FORMAT(14A4,A3,6X,I3,2XI2)                                        GRPB  0300
      IF (TYPE .NE. 99) GO TO 301                                       GRPB  0310
      ERROR = 0                                                         GRPB  0320
      RETURN                                                            GRPB  0330
  301 IF (TYPE .NE. 31) GO TO 300                                       GRPB  0340
      READ(5,101) PRINTS,SYMFCT,IORIEN,IFACT,XSC,YSC,ZSC,DELX,DELY,DELZ, GRPB  0350
```

```
DECK GRPB

    1      ISTAT3,ITAPE,IREW8,TYPE
 101  FORMAT (I1,1XI1,2I1,1X3F6.0,1X3F6.0,16XI2,2I1,7XI2)
      IF (TYPE .NE. 32) GO TO 300
      IF (IREW8 .EQ. 0) REWIND 8
      IF (SYMFCT .EQ. 0) SYMFCT = 2
      IA = 0
      IB = 0
      IF (IORIEN .EQ. 2) IB = 1
      IF (IORIEN .EQ. 3) IA = 1
    3 IF (ITAPE.EQ.0) READ (5,1) X,Y,Z,STAT,XX,YY,ZZ,STATT,TYPE
      IF (ITAPE.NE.0) READ (8,1) X,Y,Z,STAT,XX,YY,ZZ,STATT,TYPE
    1 FORMAT (2(3F10.0,I1),8XI2 )
      IF (TYPE.NE.3 .AND. ITAPE.EQ.0) GO TO 300
      IF (TYPE.NE.3 .AND. ITAPE.NE.0) GO TO 806
C
      IF (IFACT.EQ.0) GO TO 10
        X = X * XSC + DELX
        XX= XX* XSC + DELX
        Y = Y * YSC + DELY
        YY= YY* YSC + DELY
        Z = Z * ZSC + DELZ
        ZZ= ZZ* ZSC + DELZ
   10 IF (STAT.EQ.3 .OR. STATT.EQ.3 .AND. ISTAT3.GT.0) ISTAT3 = ISTAT3-1
      IF (STAT.EQ.3 .AND. ISTAT3.GT.0) STAT = 0
      IF (STATT.EQ.3 .AND. ISTAT3.GT.0) STATT = 0
      WRITE (3) X,Y,Z,STAT,XX,YY,ZZ,STATT
      IF (STAT.EQ.3 .OR. STATT.EQ.3) GO TO 4
      GO TO 3
    4 WRITE (6,4016) IPIC
 4016 FORMAT (1H1,////,1H0,34HPICTURE DRAWING PROGRAM    PICTURE
    1     7H NUMBER ,I4)
       IPIC = IPIC + 1
C
C  READ PLOTTING INSTRUCTIONS
      READ (5,5) PSI,THETA,PHI,ICS,IREFL,ISHAD,IAREA,IQUAD,IFRAME,NCAM,
     1  MARKPT,NG,MG,IG,JG,NXG,NYG,LAST,TYPE
```

```
      DECK GRPB

    5 FORMAT (F6.0,1XF6.0,1XF6.0,6(1XI1),2(1XI2),2X2I3,1X2I3,1X2I2,      GRPB 0720
     1  1XI1,10XI2 )                                                     GRPB 0730
      IF (TYPE .NE. 34) GO TO 300                                        GRPB 0740
      READ (5,6) XLG,XRG,YBG,YTG,DXG,DYG,NOSCAL,TYPE                      GRPB 0750
    6 FORMAT (5F10.0,F9.0,I1,10XI2)                                      GRPB 0760
      IF (TYPE .NE. 35) GO TO 300                                        GRPB 0770
      IF (NOSCAL .EQ. 1) GO TO 8                                         GRPB 0780
      READ (5,7) (VTITLE(I),I=1,8),(HTITLE(I),I=1,8),TYPE                GRPB 0790
    7 FORMAT (7A4,A2,7A4,A1,11XI2)                                       GRPB 0800
      IF (TYPE .NE. 36) GO TO 300                                        GRPB 0810
C                                                                        GRPB 0820
    8 CALL CAMRAV (NCAM)                                                 GRPB 0830
      CALL FRAMEV (0)                                                    GRPB 0840
C                                                                        GRPB 0850
C     SET UP STARTING CONSTANTS                                          GRPB 0860
      IFADV = 1                                                          GRPB 0870
      ISTART = 0                                                         GRPB 0880
      PSI   =  PSI   / 57.2957795                                        GRPB 0890
      THETA = THETA / 57.2957795                                         GRPB 0900
      PHI   =  PHI   / 57.2957795                                        GRPB 0910
      SINTH = SIN(THETA)                                                 GRPB 0920
      COSTH = COS(THETA)                                                 GRPB 0930
      SINPSI= SIN(PSI)                                                   GRPB 0940
      COSPSI= COS(PSI)                                                   GRPB 0950
      SINPHI= SIN(PHI)                                                   GRPB 0960
      COSPHI= COS(PHI)                                                   GRPB 0970
      A1 = COSTH * SINPSI                                                GRPB 0980
      A2 = COSPSI*COSPHI + SINTH*SINPSI*SINPHI                           GRPB 0990
      A3 =-COSPSI*SINPHI + SINTH*SINPSI*COSPHI                           GRPB 1000
      A4 =-SINTH                                                         GRPB 1010
      A5 = COSTH*SINPHI                                                  GRPB 1020
      A6 = COSTH*COSPHI                                                  GRPB 1030
      A7 = COSTH*COSPSI                                                  GRPB 1040
      A8 =-SINPSI*COSPHI + SINTH*COSPSI*SINPHI                           GRPB 1050
      A9 = SINPSI*SINPHI + SINTH*COSPSI*COSPHI                           GRPB 1060
       N = -1                                                            GRPB 1070
```

```
DECK GRPB
      NN = - 1                                                          GRPB 1080
       KLCT = 0                                                         GRPB 1090
       L = 0                                                            GRPB 1100
       NPRT = 10                                                        GRPB 1110
       AREAT = 0.0                                                      GRPB 1120
       VOL = 0.0                                                        GRPB 1130
      REWIND 3                                                          GRPB 1140
C                                                                       GRPB 1150
C                                                                       GRPB 1160
C     READ IN ALL SURFACE DATA                                          GRPB 1170
   29 READ (3) X,Y,Z,STAT, XX,YY,ZZ,STATT                               GRPB 1180
       RFLAG = .FALSE.                                                  GRPB 1190
       GO TO 80                                                         GRPB 1200
   30 IF (RFLAG) GO TO 50                                               GRPB 1210
       RFLAG = .TRUE.                                                   GRPB 1220
       X = XX                                                           GRPB 1230
       Y = YY                                                           GRPB 1240
       Z = ZZ                                                           GRPB 1250
       STAT = STATT                                                     GRPB 1260
       GO TO 60                                                         GRPB 1270
   50 RFLAG = .FALSE.                                                   GRPB 1280
      READ (3) X,Y,Z,STAT, XX,YY,ZZ,STATT                               GRPB 1290
   60 IF (STAT .EQ. 0 .OR. STAT .EQ. 3) GO TO 180                       GRPB 1300
      IF (STAT .EQ. 2) GO TO 200                                        GRPB 1310
   70 IF (.NOT. AFLAG) GO TO 200                                        GRPB 1320
       MC = M                                                           GRPB 1330
   80 M = 1                                                             GRPB 1340
       IF (STAT .EQ. 2) GO TO 150                                       GRPB 1350
       IF (.NOT. BFLAG) GO TO 84                                        GRPB 1360
   75 DO 81 J =1,MC                                                     GRPB 1370
        XA(J) = XB(J)                                                   GRPB 1380
        YA(J) = YB(J)                                                   GRPB 1390
   81   ZA(J) = ZB(J)                                                   GRPB 1400
   83 XB(1) = X                                                         GRPB 1410
      YB(1) = Y                                                         GRPB 1420
      ZB(1) = Z                                                         GRPB 1430
```

DECK GRPB

```
      GO TO 30
   84 IF (AFLAG) GO TO 85
       BFLAG = .TRUE.
       GO TO 75
   85 AFLAG = .FALSE.
       GO TO 83
  150 AFLAG = .TRUE.
       BFLAG = .FALSE.
       N = N+1
  160 XA(M) = X
      YA(M) = Y
      ZA(M) = Z
      GO TO 30
  180 M = M + 1
      IF (AFLAG) GO TO 160
       XB(M) = X
       YB(M) = Y
       ZB(M) = Z
      IF (STAT .NE. 3) GO TO 30
  200 MMIN = MINO (M,MC) - 1
       NN2 = 1
       MC = M
  250 N = N + 1
      NN = NN + 1
       KLCT = KLCT + 1
C
C
C  BEGIN COMPUTATION OF SURFACE ELEMENT CHARACTERISTICS
C
  450 DO 2000 I= 1,MMIN
      IIA = I + IA
      IIB = I + IB
        XIN(1) = XA(IIA )
        XIN(2) = XA(IIA +1)
        XIN(3) = XB(IIB +1)
        XIN(4) = XB(IIB )
```

```
DECK GRPB
         YIN(1) = YA(IIA )
         YIN(2) = YA(IIA +1)
         YIN(3) = YB(IIB +1)
         YIN(4) = YB(IIB )
         ZIN(1) = ZA(IIA )
         ZIN(2) = ZA(IIA +1)
         ZIN(3) = ZB(IIB +1)
         ZIN(4) = ZB(IIB )
         IRFLG = 0
C
C  FORM DIAGONAL VECTORS
       T1X = XIN(3) - XIN(1)
       T2X = XIN(4) - XIN(2)
       T1Y = YIN(3) - YIN(1)
       T2Y = YIN(4) - YIN(2)
       T1Z = ZIN(3) - ZIN(1)
       T2Z = ZIN(4) - ZIN(2)
C
C  FORM CROSS PRODUCT  N=T2 X T1
       NX = T2Y*T1Z - T1Y*T2Z
       NY = T1X*T2Z - T2X*T1Z
       NZ = T2X*T1Y - T1X*T2Y
       VN = SQRT ( NX*NX + NY*NY + NZ*NZ )
C
C  FORM UNIT NORMAL VECTOR
      IF (VN .EQ. 0.0) GO TO 421
      NX = NX / VN
       NY = NY / VN
       NZ = NZ / VN
C
C  COMPUTE AVERAGE POINT
 421  AVX = 0.25 * (XIN(1) + XIN(2) + XIN(3) + XIN(4) )
       AVY = 0.25 * (YIN(1) + YIN(2) + YIN(3) + YIN(4) )
       AVZ = 0.25 * (ZIN(1) + ZIN(2) + ZIN(3) + ZIN(4) )
C
C  COMPUTE PROJECTION DISTANCE
```

DECK GRPB

```
      D = NX*(AVX - XIN(1)) + NY*(AVY - YIN(1)) + NZ*(AVZ-ZIN(1))     GRPB 2160
      PD = ABS(D)                                                       GRPB 2170
C                                                                       GRPB 2180
C                                                                       GRPB 2190
      T = SQRT (T1X*T1X + T1Y*T1Y + T1Z*T1Z)                            GRPB 2200
     IF (T .EQ. 0.0) GO TO 431                                          GRPB 2210
     T1X = T1X / T                                                      GRPB 2220
      T1Y = T1Y / T                                                     GRPB 2230
      T1Z = T1Z / T                                                     GRPB 2240
C                                                                       GRPB 2250
C                                                                       GRPB 2260
 431 T2X = NY*T1Z - NZ*T1Y                                              GRPB 2270
      T2Y = NZ*T1X - NX*T1Z                                             GRPB 2280
      T2Z = NX*T1Y - NY*T1X                                             GRPB 2290
C                                                                       GRPB 2300
C  COMPUTE COORDINATES OF CORNER POINTS IN REFERENCE COORD. SYSTEM      GRPB 2310
C                                                                       GRPB 2320
     DO 1000 J = 1,4                                                    GRPB 2330
      XPA= XIN(J) + NX*D                                                GRPB 2340
      YPA= YIN(J) + NY*D                                                GRPB 2350
      ZPA= ZIN(J) + NZ*D                                                GRPB 2360
C                                                                       GRPB 2370
     IF (IQUAD .EQ. 0) GO TO 470                                        GRPB 2380
       XIN(J) = XPA                                                     GRPB 2390
       YIN(J) = YPA                                                     GRPB 2400
       ZIN(J) = ZPA                                                     GRPB 2410
C                                                                       GRPB 2420
 470  D = - D                                                           GRPB 2430
      XDIF = XPA- AVX                                                   GRPB 2440
      YDIF = YPA- AVY                                                   GRPB 2450
      ZDIF = ZPA- AVZ                                                   GRPB 2460
C                                                                       GRPB 2470
C  TRANSFORM CORNER POINTS TO ELEMENT COORDINATE SYSTEM (XI,ETA) WITH   GRPB 2480
C   AVERAGE POINT AS ORIGIN                                             GRPB 2490
      XI(J) = T1X*XDIF + T1Y*YDIF + T1Z*ZDIF                            GRPB 2500
 1000 ETA(J)= T2X*XDIF + T2Y*YDIF + T2Z*ZDIF                            GRPB 2510
```

DECK GRPB

```
C                                                                       GRPB  2520
C  COMPUTE CENTROID                                                     GRPB  2530
      ETACK = ETA(2) - ETA(4)                                           GRPB  2540
      IF (ETACK .NE. 0.0) GO TO 432                                     GRPB  2550
       XIO = 0.0                                                        GRPB  2560
       GO TO 433                                                        GRPB  2570
 432  XIO = .333333333 * (XI(4) * (ETA(1)-ETA(2)) + XI(2)               GRPB  2580
     1      * (ETA(4)-ETA(1))) / (ETA(2)-ETA(4))                        GRPB  2590
 433  ETAO = -.333333333 * ETA(1)                                       GRPB  2600
C                                                                       GRPB  2610
C  OBTAIN CORNER POINTS IN SYSTEM WITH CENTROID AS ORIGIN               GRPB  2620
       DO 1020 J = 1,4                                                  GRPB  2630
        XI(J) = XI(J) - XIO                                             GRPB  2640
 1020   ETA(J) = ETA(J) - ETAO                                          GRPB  2650
C                                                                       GRPB  2660
C  TRANSFORM CENTROID TO REFERENCE COORDINATE SYSTEM                    GRPB  2670
       XCENT = AVX + T1X*XIO + T2X*ETAO                                 GRPB  2680
       YCENT = AVY + T1Y*XIO + T2Y*ETAO                                 GRPB  2690
       ZCENT = AVZ + T1Z*XIO + T2Z*ETAO                                 GRPB  2700
C                                                                       GRPB  2710
C  CONSTANTS                                                            GRPB  2720
       XI3M1 = XI(3) - XI(1)                                            GRPB  2730
       ETA2M4 = ETA(2) - ETA(4)                                         GRPB  2740
C                                                                       GRPB  2750
C  COMPUTE AREA AND VOLUME OF ELEMENTS                                  GRPB  2760
       AREA = 0.5 * XI3M1 * ETA2M4                                      GRPB  2770
      AREAT = AREAT + AREA                                              GRPB  2780
       DELVOL = AREA * NY * YCENT                                       GRPB  2790
      VOL = VOL + DELVOL                                                GRPB  2800
      L = L + 1                                                         GRPB  2810
      IF (PRINTS.EQ.0) GO TO 1770                                       GRPB  2820
C  PRINT RESULTS OF CALCULATIONS TO DETERMINE ELEMENT CHARACTERISTICS   GRPB  2830
 1700 IF (NPRT.GE.9) GO TO 1750                                         GRPB  2840
       NPRT = NPRT + 1                                                  GRPB  2850
      IF (I .EQ. 1) GO TO 1760                                          GRPB  2860
       WRITE (6,4005) I, XIN, NX, XCENT, AREA,L,YIN,NY,YCENT,DELVOL,ZIN, GRPB  2870
```

DECK GRPB

```
    1  NZ,ZCENT,VOL
       GO TO 1770
 1750 NPRT = 0
      CALL HEADR2
      WRITE (6,4002)
 1760 WRITE (6, 4010) N, I, XIN, NX, XCENT, AREA,L,YIN,NY,YCENT,DELVOL,
    1 ZIN,NZ,ZCENT,VOL
 1770 IF (AREA .LT. 0.1E-09) GO TO 2000
C
C  CHECK IF NEW GRID IS REQUIRED AND PREPARE GRID
      IF (IFADV .EQ. 0) GO TO 471
C
      IF (NOSCAL .EQ. 0) GO TO 505
       CALL STOPTV
       CALL BRITEV
       CALL XSCALV (XLG,XRG,24,0)
       CALL YSCALV (YBG,YTG,0,24)
       GO TO 511
 505  CALL GRID1V (2,XLG,XRG,YBG,YTG,DXG,DYG,NG,MG,IG,JG,NXG,NYG)
      DO 510 II=1,3
 510  CALL APRNTV (0,-12,30,VTITLE,8,689)
      DO 520 II=1,3
 520  CALL PRINTV (29,HTITLE,391,8)
 511  IF (IFRAME.EQ.1 .AND. ISTART.EQ.1) GO TO 521
      READ (5,522) (HLABEL(II),II=1,15),TYPE
 522  FORMAT(14A4,1A3,11XI2)
      IF (TYPE .NE. 37) GO TO 300
      ISTART = 1
 521  DO 523 II=1,3
      CALL PRINTV (-45,45HHYPERSONIC ARBITRARY-BODY AERODYNAMIC PROGRAM,
    1  330,1023)
 523  CALL PRINTV (59,HLABEL,248,1007)
      IF (IFRAME .NE. 1) GO TO 525
      CALL SCSETV (4)
      WRITE (16,524) XIN(1),XIN(4)
 524  FORMAT (1H ,10X11HSTATIONS  =F9.3,8H   AND =F9.3 )
```

```
DECK GRPB
 525   IFADV = 0
C
 471   NXO = THIRD(NX,NY,NZ,PSI,THETA,PHI)
       IF (NXO.LE.0.0 .AND. ISHAD.EQ.0) GO TO 571
C
C
C  CALCULATE POINTS TO BE PLOTTED
 530   YO1 = FIRST(XIN(1),YIN(1),ZIN(1),A1,A2,A3)
       YO2 = FIRST(XIN(2),YIN(2),ZIN(2),A1,A2,A3)
       YO3 = FIRST(XIN(3),YIN(3),ZIN(3),A1,A2,A3)
       YO4 = FIRST(XIN(4),YIN(4),ZIN(4),A1,A2,A3)
       ZO1 = FIRST(XIN(1),YIN(1),ZIN(1),A4,A5,A6)
       ZO2 = FIRST(XIN(2),YIN(2),ZIN(2),A4,A5,A6)
       ZO3 = FIRST(XIN(3),YIN(3),ZIN(3),A4,A5,A6)
       ZO4 = FIRST(XIN(4),YIN(4),ZIN(4),A4,A5,A6)
C
       YIN2(1) = YO1
       YIN2(2) = YO2
       YIN2(3) = YO3
       YIN2(4) = YO4
       ZIN2(1) = ZO1
       ZIN2(2) = ZO2
       ZIN2(3) = ZO3
       ZIN2(4) = ZO4
C
       CALL APLOTV (4,YIN2,ZIN2,1,1,1,MARKPT,IERR)
C
       IF (ICS .EQ. 3) GO TO 571
C
       IF (ICS.EQ.0 .OR. ICS.EQ.1) GO TO 540
        GO TO 541
 540    IX1 = NXV(YO1)
        IY1 = NYV(ZO1)
       CALL SCERRV (KX,KY)
        IX2 = NXV(YO2)
        IY2 = NYV(ZO2)
```

DECK GRPB

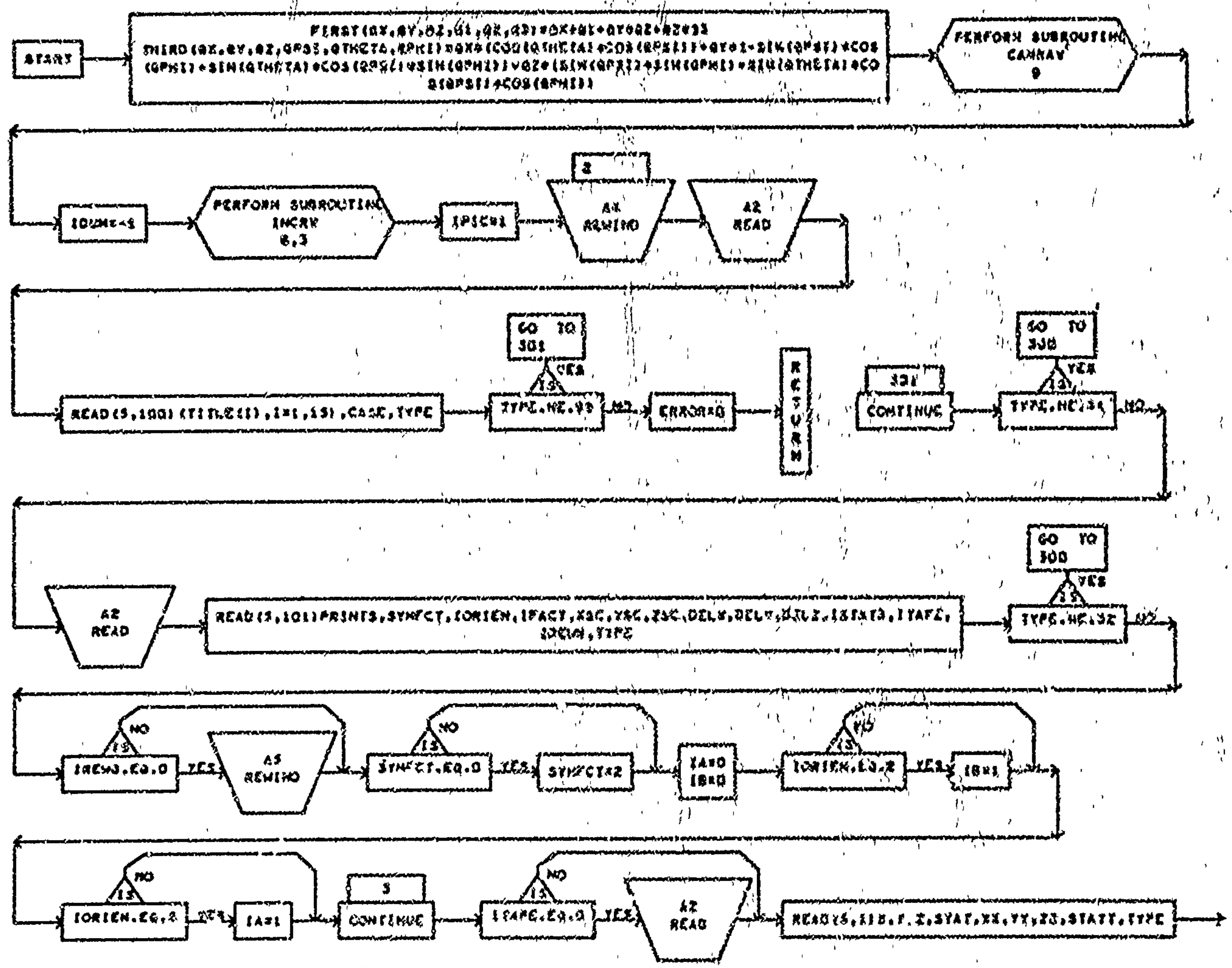
```
      CALL  SCERRV (KX1,KY1)
      KKXY = KX+KY+KX1+KY1
      IF (KKXY.NE.0) GO TO 541
      IF (NXO.GT.0.0) CALL LINEV (IX1,IY1,IX2,IY2)
      IF (NXO.LE.0.0) CALL DOTLNV(IX1,IY1,IX2,IY2)
C
  541 IF (ICS.EQ.0 .OR. ICS.EQ.2 .OR. ICS.EQ.4) GO TO 550
       GO TO 551
  550  IX1 = NXV(YO2)
       IY1 = NYV(ZO2)
      CALL  SCERRV (KX,KY)
       IX2 = NXV(YO3)
       IY2 = NYV(ZO3)
      CALL  SCERRV (KX1,KY1)
      KKXY = KX+KY+KX1+KY1
      IF (KKXY.NE.0) GO TO 551
      IF (NXO.GT.0.0) CALL LINEV (IX1,IY1,IX2,IY2)
      IF (NXO.LE.0.0) CALL DOTLNV(IX1,IY1,IX2,IY2)
C
  551 IF (ICS.EQ.0 .OR. ICS.EQ.1 .OR. ICS.EQ.4) GO TO 560
       GO TO 561
  560  IX1 = NXV(YO3)
       IY1 = NYV(ZO3)
      CALL  SCERRV (KX,KY)
       IX2 = NXV(YO4)
       IY2 = NYV(ZO4)
      CALL  SCERRV (KX1,KY1)
      KKXY = KX+KY+KX1+KY1
      IF (KKXY.NE.0) GO TO 561
      IF (NXO.GT.0.0) CALL LINEV (IX1,IY1,IX2,IY2)
      IF (NXO.LE.0.0) CALL DOTLNV(IX1,IY1,IX2,IY2)
C
  561 IF (ICS.EQ.0 .OR. ICS.EQ.2) GO TO 570
       GO TO 571
  570  IX1 = NXV(YO1)
       IY1 = NYV(ZO1)
```

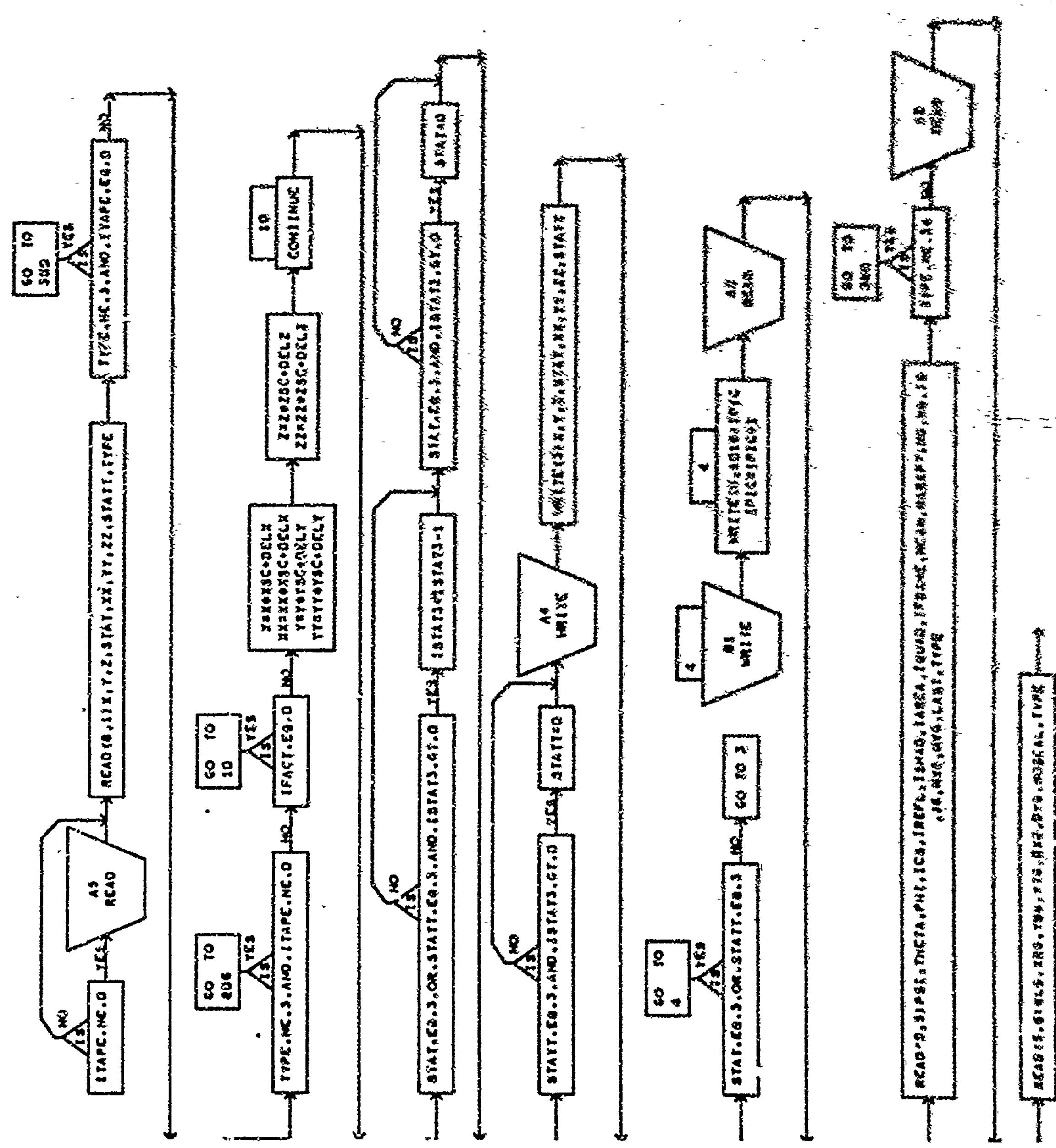
```
DECK GRPB

      CALL SCERRV (KX,KY)
       IX2 = NXV(YO4)
       IY2 = NYV(ZO4)
      CALL SCERRV (KX1,KY1)
      KKXY = KX+KY+KX1+KY1
      IF (KKXY.NE.0) GO TO 571
      IF (NXO.GT.0.0) CALL LINEV (IX1,IY1,IX2,IY2)
      IF (NXO.LE.0.0) CALL DOTLNV(IX1,IY1,IX2,IY2)
C
 571  IF (IREFL .EQ. 0 .OR. IRFLG .EQ. 3) GO TO 2000
      IF (IREFL .EQ. 2 .AND. IRFLG .EQ. 1) GO TO 600
      IF (IREFL .EQ. 2 .AND. IRFLG .EQ. 2) GO TO 602
C
C     REFLECT QUADRANT I ELEMENTS TO QUADRANT II
      DO 580 II = 1,4
 580   YIN(II) = -YIN(II)
       NY = -NY
       GO TO 604
C
C     REFLECT QUADRANT II ELEMENTS TO QUADRANT IV
 600  DO 601 II = 1,4
       YIN(II) = -YIN(II)
 601   ZIN(II) = -ZIN(II)
       NY = -NY
       NZ = -NZ
       GO TO 604
C
C     REFLECT QUADRANT IV ELEMENTS TO QUADRANT III
 602  DO 603 II = 1,4
 603   YIN(II) = -YIN(II)
       NY = -NY
C
C
 604  IRFLG = IRFLG + 1
      IF (IREFL .EQ. 1) IRFLG = 3
      GO TO 471
```

```
DECK  GRPB
C                                                                        GRPB  4320
C                                                                        GRPB  4330
 2000 CONTINUE                                                           GRPB  4340
 2001 IF (STAT .LT. 2) GO TO 480                                         GRPB  4350
       NPRT = NPRT + 1                                                   GRPB  4360
      WRITE (6,472) AREAT,L,VOL                                          GRPB  4370
      NN = NN - 1                                                        GRPB  4380
      N = - 1                                                            GRPB  4390
      IF (IAREA .EQ. 0) GO TO 475                                        GRPB  4400
       CALL SCSETV (3)                                                   GRPB  4410
       WRITE (16,472) AREAT,L                                            GRPB  4420
 472   FORMAT( 1H ,25X31HTOTAL AREA OF INPUT ELEMENTS   = F14.4,         GRPB  4430
     1        6X26HTOTAL NUMBER OF ELEMENTS = I5/1H ,23X,                GRPB  4440
     2 33H TOTAL VOLUME OF INPUT ELEMENTS =F12.3)                        GRPB  4450
 475  IF (IFRAME .EQ. 2) IFADV = 1                                       GRPB  4460
 480  IF (IFRAME .EQ. 1) IFADV = 1                                       GRPB  4470
 485  IF (IFADV .EQ. 1) CALL FRAMEV (0)                                  GRPB  4480
C                                                                        GRPB  4490
C                                                                        GRPB  4500
C  TEST FOR END OF CASE                                                  GRPB  4510
 2020 IF (STAT .NE. 3) GO TO 80                                          GRPB  4520
      IF (LAST .EQ. 1) GO TO 2                                           GRPB  4530
      PRINTS=0                                                           GRPB  4540
      GO TO 4                                                            GRPB  4550
C                                                                        GRPB  4560
C                                                                        GRPB  4570
C  ERROR CHECK ON READING CARDS                                          GRPB  4580
 300  WRITE (6,4003)                                                     GRPB  4590
C                                                                        GRPB  4600
 4003 FORMAT (1H0,47H*****YOU HAVE MADE AN ERROR EITHER IN CARD TYPE     GRPB  4610
     1 49H INDICATION OR CARD ORDER - CHECK YOUR CARDS*****  )           GRPB  4620
      READ (5,810) (CARD(II),II=1,20)                                    GRPB  4630
 810  FORMAT (20A4)                                                      GRPB  4640
      WRITE (6,805) (CARD(II),II=1,20)                                   GRPB  4650
 805  FORMAT (1H0,45H THE CARD LOCATED JUST BEFORE THE CARD LISTED       GRPB  4660
     1 18H BELOW IS IN ERROR,/1H ,10X,20A4)                              GRPB  4670
```

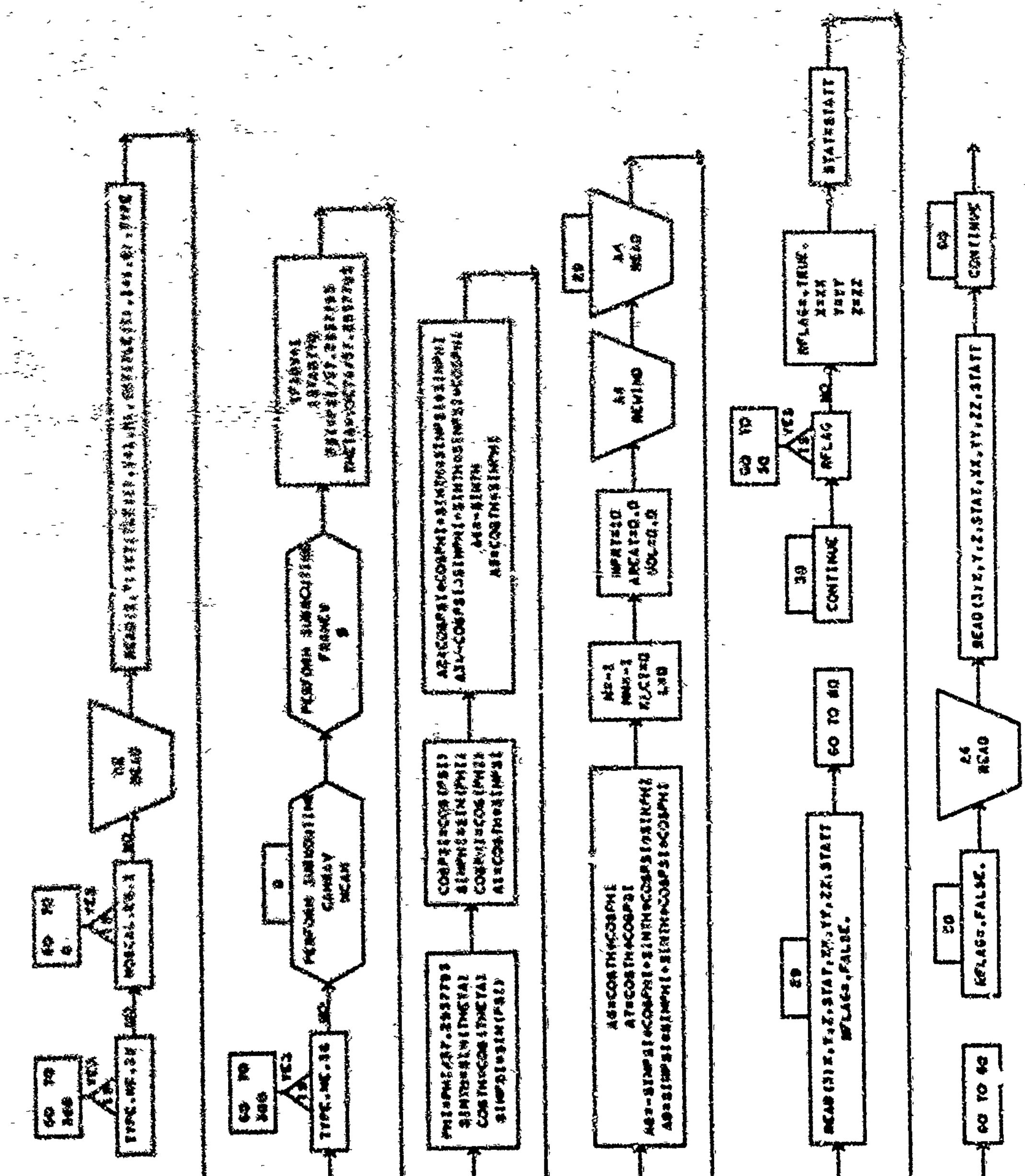
```
DECK GRPB

      GO TO 807                                                         GRPB  4680
  806 BACKSPACE 8                                                       GRPB  4690
      READ (8,810) (CARD(II),II=1,20)                                   GRPB  4700
      WRITE (6,808) (CARD(II),II=1,20)                                  GRPB  4710
  8C8 FORMAT (1HO,46H***THE FOLLOWING CARD ON TAPE 8 IS IN ERROR***,/   GRPB  4720
     1 1H ,10X,20A4)                                                    GRPB  4730
  807 CALL FRAMEV (0)                                                   GRPB  4740
      CALL SCSETV (4)                                                   GRPB  4750
      WRITE (16,4004)                                                   GRPB  4760
 4004 FORMAT (1H ,45HNO MORE SC-4020 DATA IS PLOTTED BECAUSE OF AN      GRPB  4770
     126H ERROR IN YOUR INPUT CARDS )                                   GRPB  4780
      ERROR = 1                                                         GRPB  4790
      RETURN                                                            GRPB  4800
C                                                                       GRPB  4810
 4002 FORMAT (1HO,28H  INPUT SURFACE ELEMENT DATA/1HO,6X1HN3X1HM7X1HX,  GRPB  4820
     1 3(13X,1HX),11X2HNX9X5HXCENT9X4HAREA8X1HL ,/1H ,5X, 4(13X,1HY),   GRPB  4830
     2 11X2HNY9X5HYCENT ,7X,7HDELTA V,/1H ,5X,4(13X,1HZ),11X2HNZ,       GRPB  4840
     3 9X,5HZCENT ,7X,6HVOLUME,/1H  )                                   GRPB  4850
 4005 FORMAT (1HO,7X, I4, 1P4E14.5,0PF10.6,1P2E14.5,I6,2(/12X,4E14.5,   GRPB  4860
     1 OPF10.6,1P2E14.5)  )                                             GRPB  4870
 4010 FORMAT (1HO,3X, 2I4,1P4E14.5,0PF10.6,1P2E14.5,I6,2(/12X,4E14.5,   GRPB  4880
     1 OPF10.6,1P2E14.5)  )                                             GRPB  4890
C                                                                       GRPB  4900
      END                                                               GRPB  4910
```

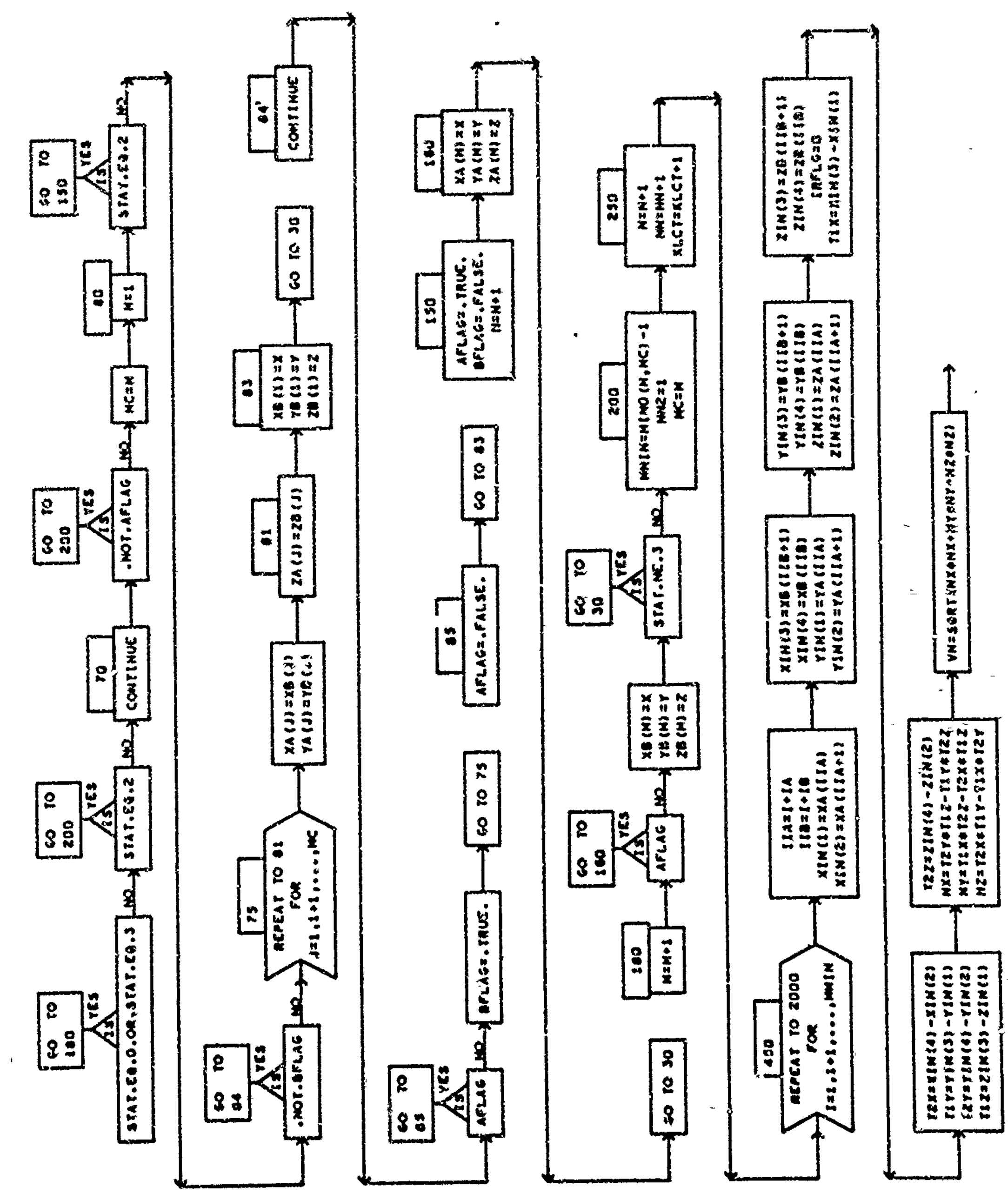
SUBROUTINE FACTOR

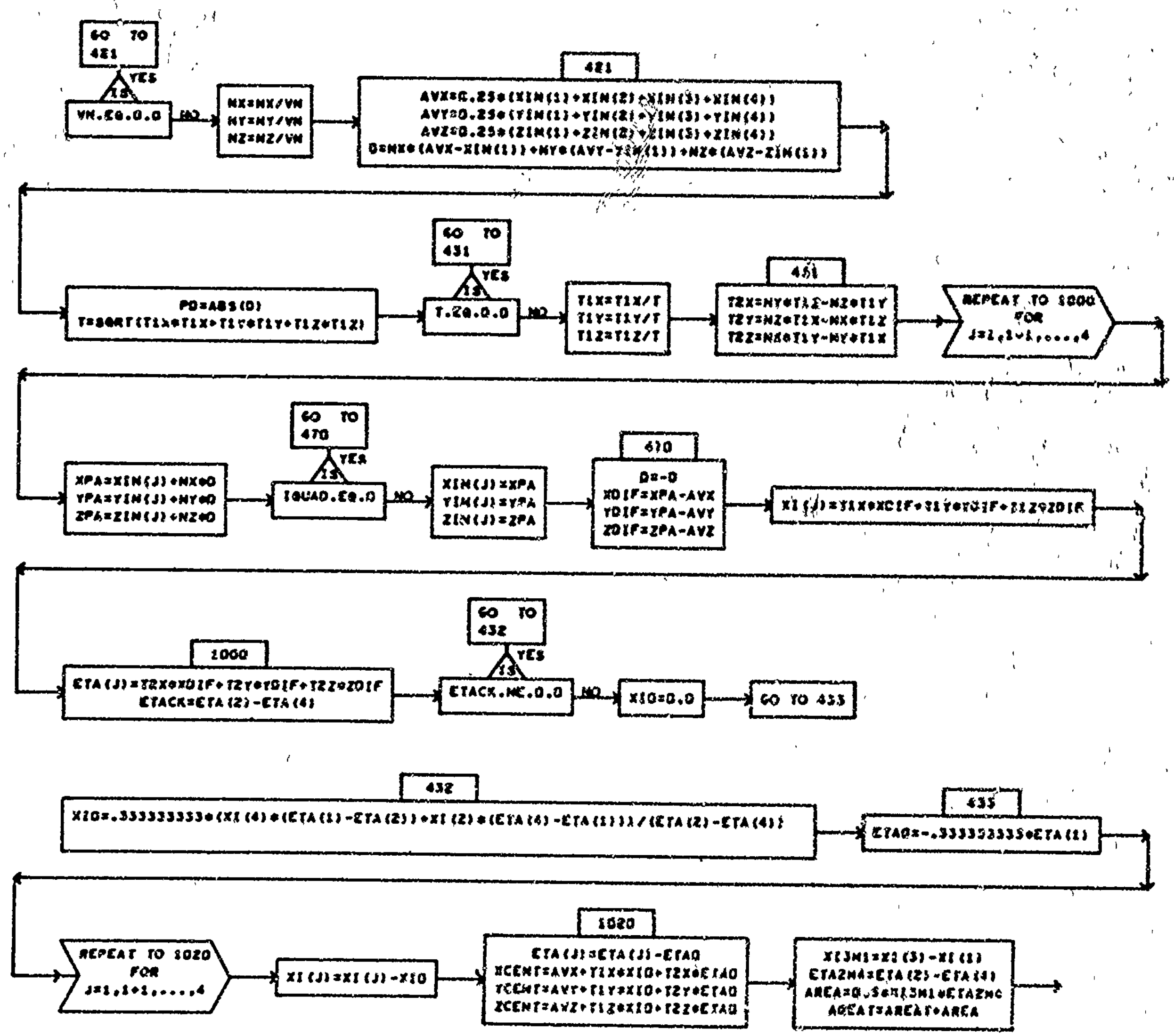


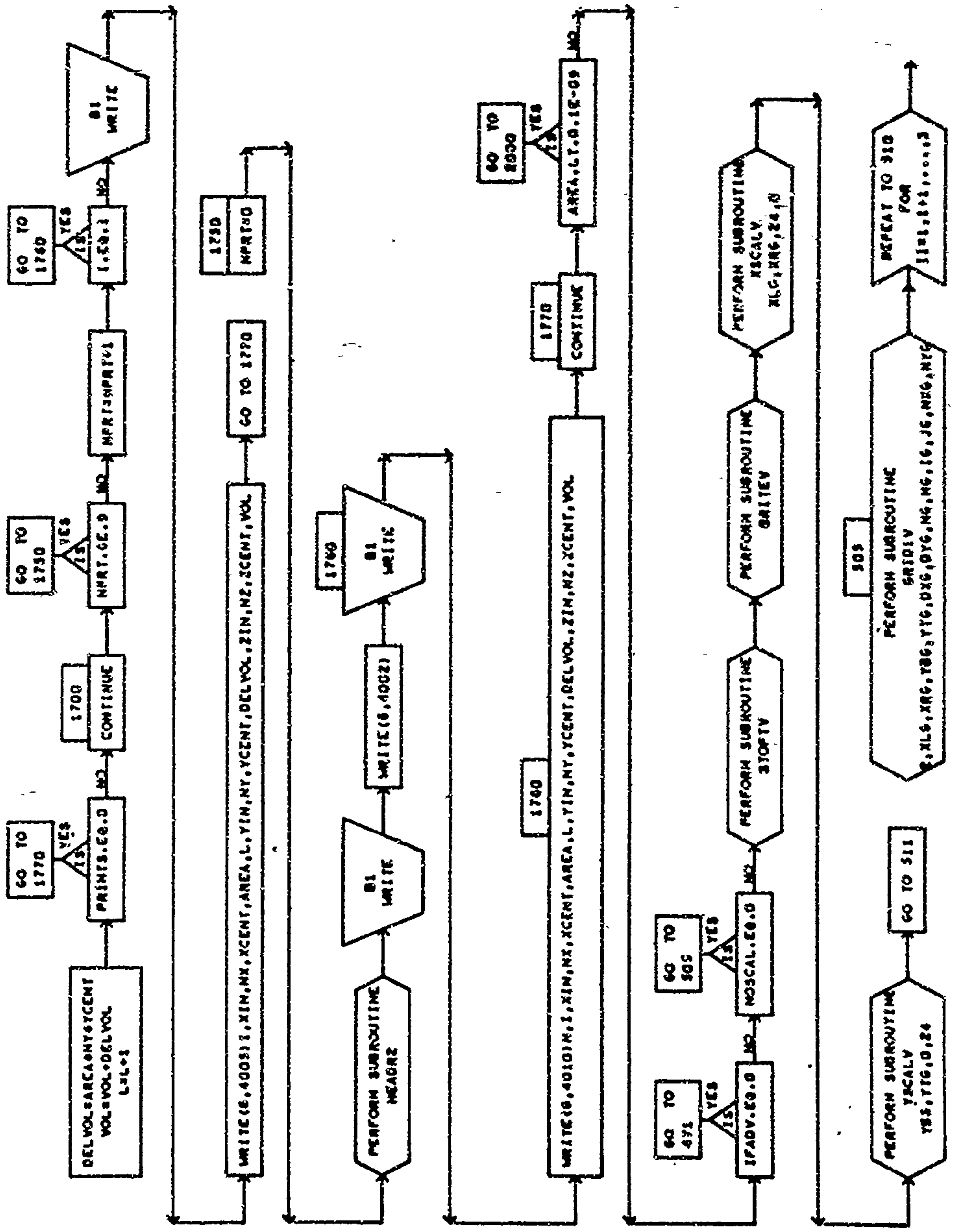


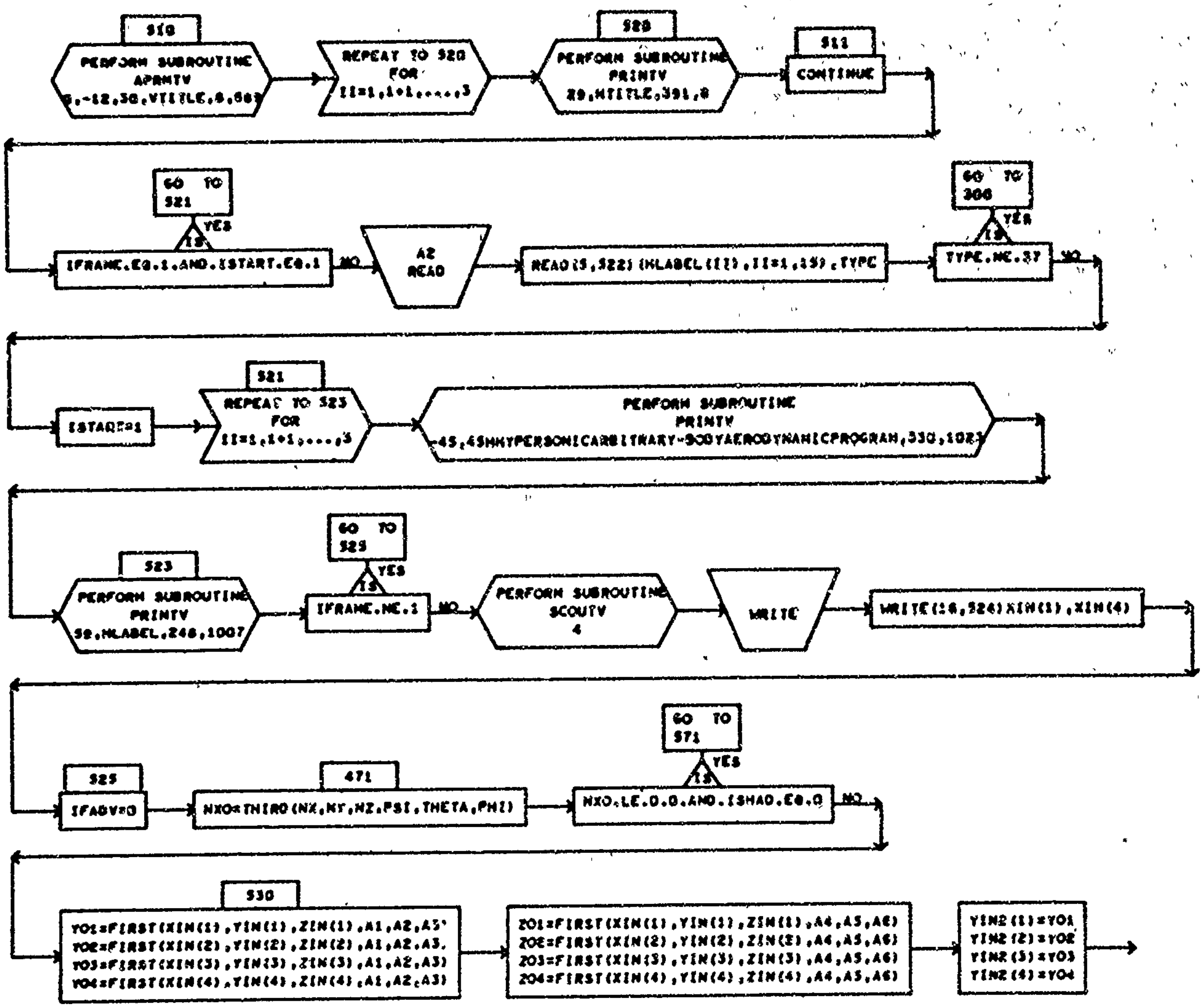
A3 READ
A4 WRITE
B1 WRITE
CONTINUE
STATT=0
GO TO 3
GO TO 10

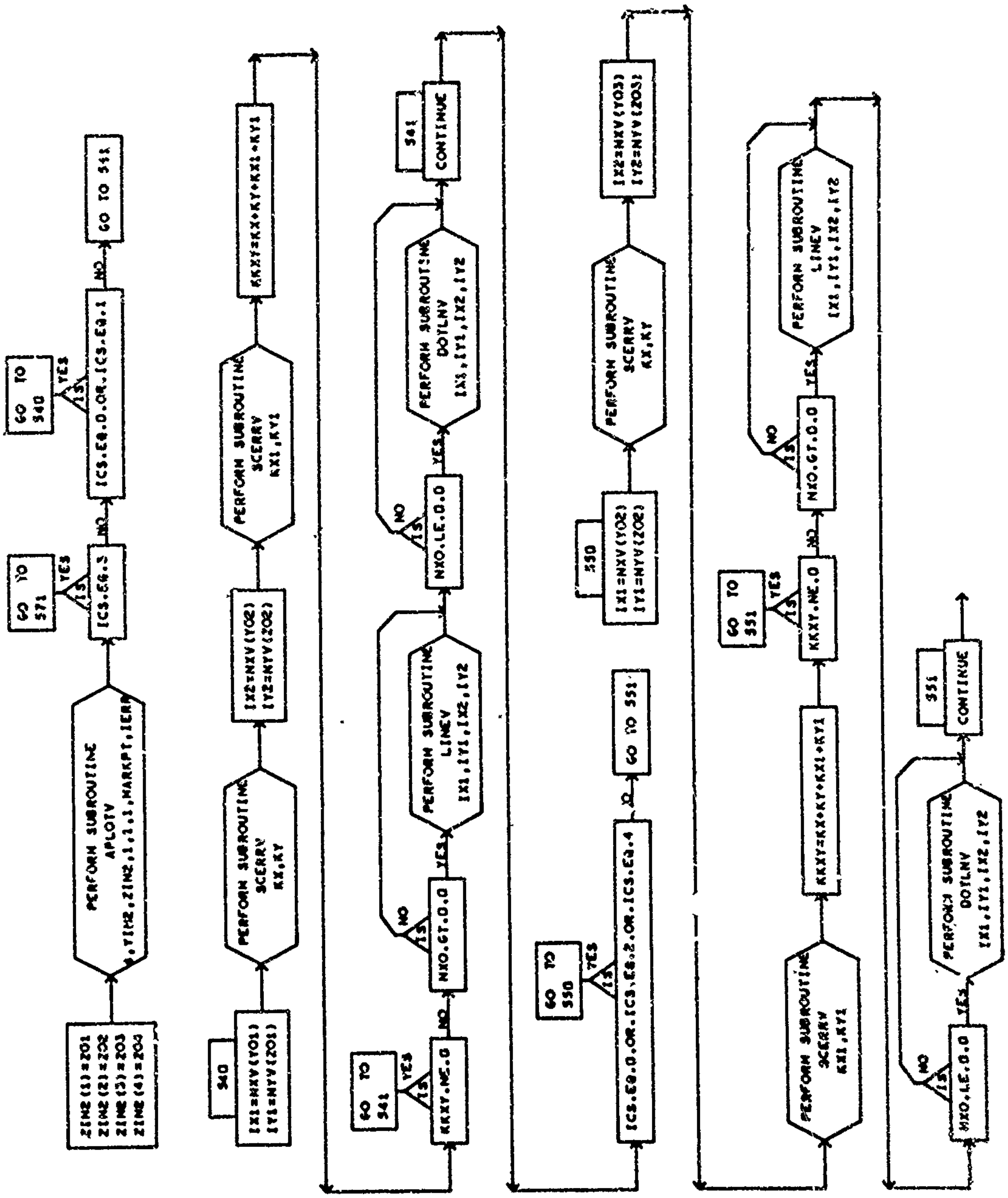


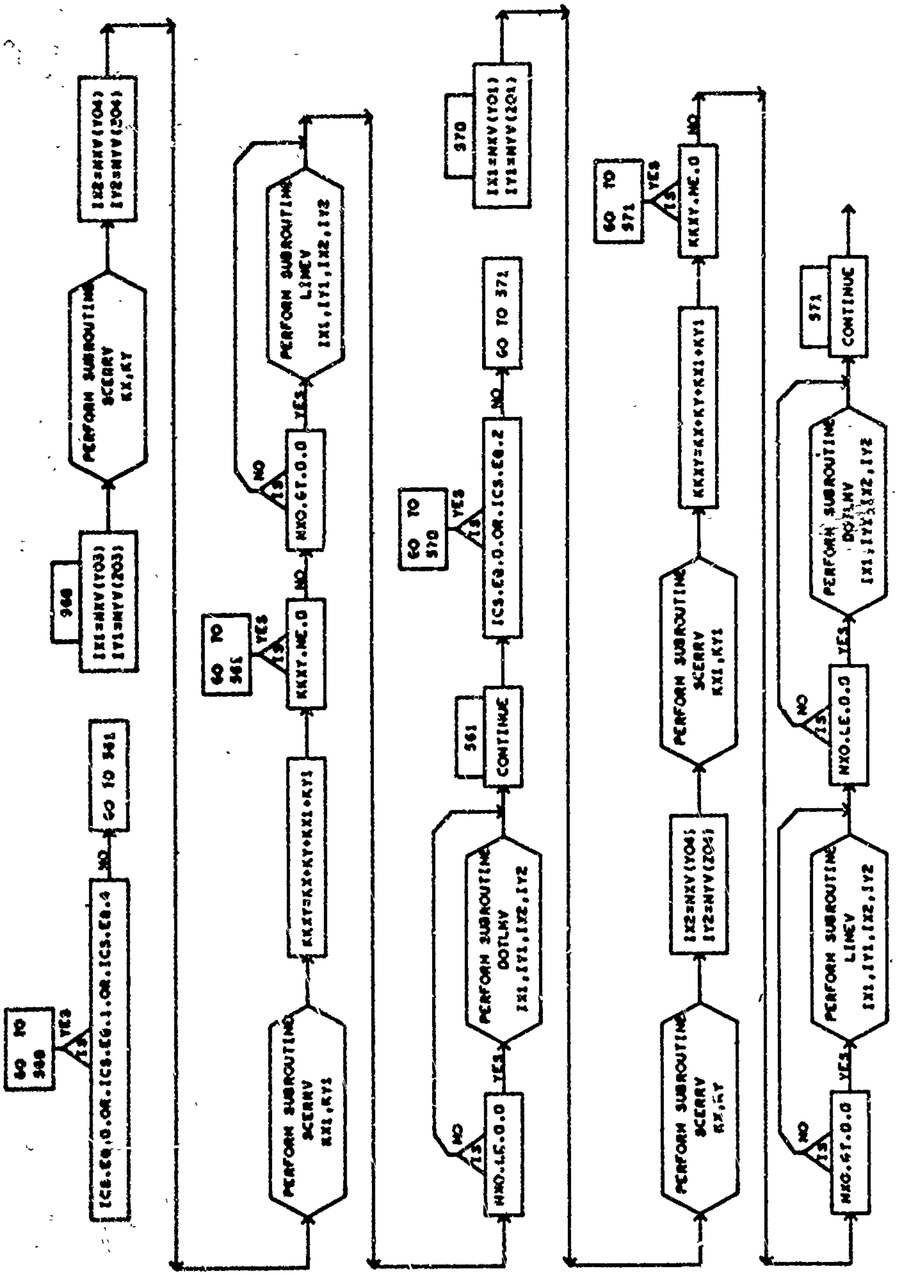

GO TO 180
STAT.EQ.0.OR.STAT.EQ.3
GO TO 200
STAT.EQ.2
70
CONTINUE
GO TO 200
.NOT.AFLAG
NC=N
80
N=1
GO TO 150
STAT.EQ.2
GO TO 84
.NOT.BFLAG
75
REPEAT TO 81 FOR J=1,1+1,....,NC
XA(J)=XB(J)
81
ZA(J)=ZB(J)
83
XB(1)=X
YB(1)=Y
ZB(1)=Z
GO TO 30
84'
CONTINUE
GO TO 85
AFLAG
BFLAG=.TRUE.
GO TO 75
85
AFLAG=.FALSE.
GO TO 83
150
AFLAG=.TRUE.
BFLAG=.FALSE.
N=N+1
180
XA(N)=X
YA(N)=Y
ZA(N)=Z
GO TO 30
180
N=N+1
GO TO 180
AFLAG
XB(N)=X
YB(N)=Y
ZB(N)=Z
GO TO 30
STAT.NE.3
200
NN2=1
NC=N
250
N=N+1
KLCT=KLCT+1
450
REPEAT TO 2000 FOR I=1,1+1,....,NMIN
IIA=I+IA
IIB=I+IB
XIN(1)=XA(IIA)
XIN(2)=XA(IIA+1)
XIN(3)=XB(IIB+1)
XIN(4)=XB(IIB)
YIN(1)=YA(IIA)
YIN(2)=YA(IIA+1)
YIN(3)=YB(IIB+1)
YIN(4)=YB(IIB)
ZIN(1)=ZA(IIA)
ZIN(2)=ZA(IIA+1)
ZIN(3)=ZB(IIB+1)
ZIN(4)=ZB(IIB)
VN=SQRT(NX*NX+NY*NY+NZ*NZ)

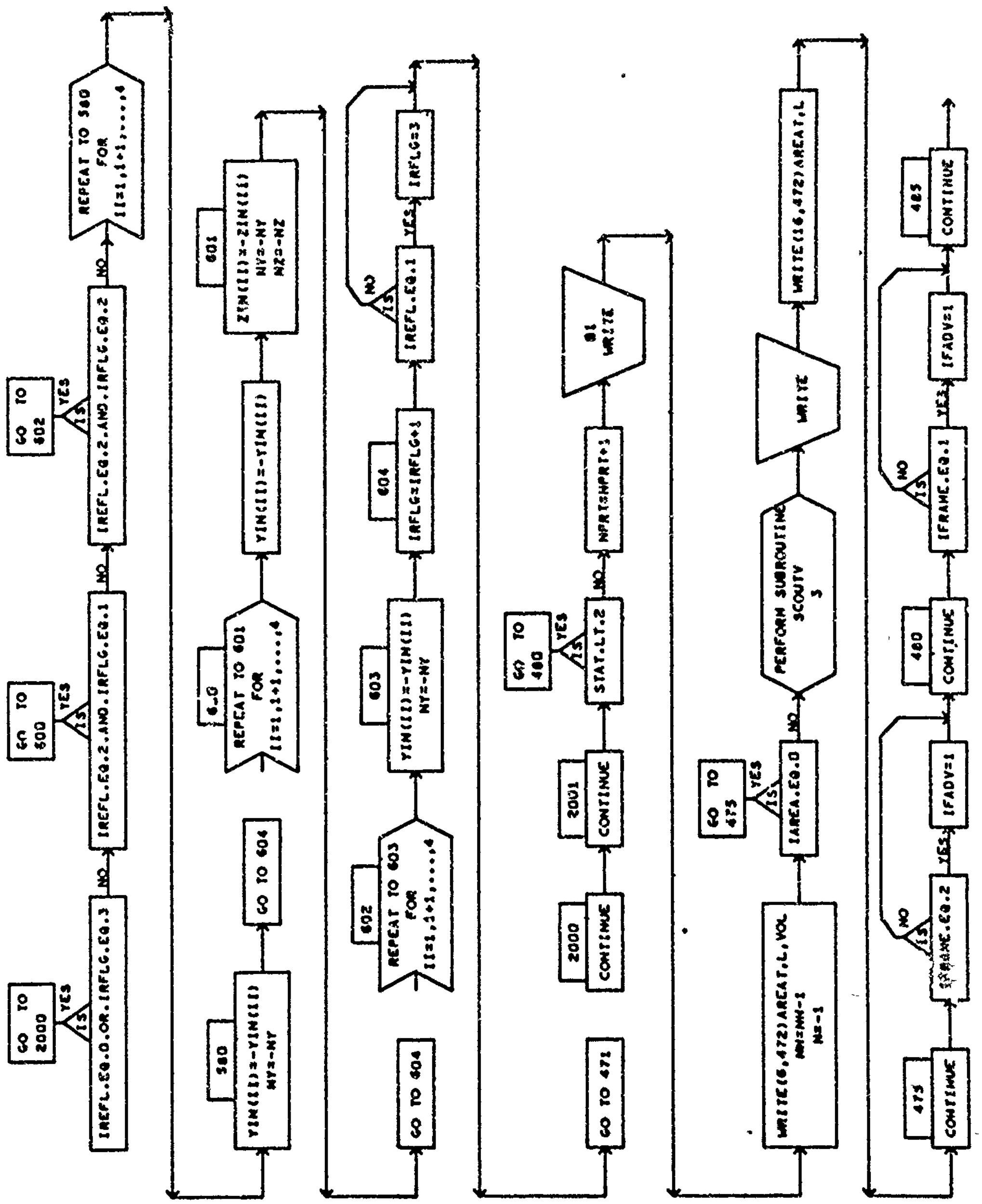

GO TO 421
YES
IS
VN.EQ.0.0
NO
NX=NX/VN
NY=NY/VN
NZ=NZ/VN
421
AVX=0.25*(XIN(1)+XIN(2)+XIN(3)+XIN(4))
AVY=0.25*(YIN(1)+YIN(2)+YIN(3)+YIN(4))
AVZ=0.25*(ZIN(1)+ZIN(2)+ZIN(3)+ZIN(4))
D=NX*(AVX-XIN(1))+NY*(AVY-YIN(1))+NZ*(AVZ-ZIN(1))
PD=ABS(D)
T=SQRT(T1X*T1X+T1Y*T1Y+T1Z*T1Z)
GO TO 431
YES
IS
T.EQ.0.0
NO
T1X=T1X/T
T1Y=T1Y/T
T1Z=T1Z/T
431
T2X=NY*T1Z-NZ*T1Y
T2Y=NZ*T1X-NX*T1Z
T2Z=NX*T1Y-NY*T1X
REPEAT TO 1000
FOR
J=1,1+1,....,4
XPA=XIN(J)+NX*D
YPA=YIN(J)+NY*D
ZPA=ZIN(J)+NZ*D
GO TO 470
YES
IS
IQUAD.EQ.0
NO
XIN(J)=XPA
YIN(J)=YPA
ZIN(J)=ZPA
470
D=-D
XDIF=XPA-AVX
YDIF=YPA-AVY
ZDIF=ZPA-AVZ
XI(J)=T1X*XDIF+T1Y*YDIF+T1Z*ZDIF
1000
ETA(J)=T2X*XDIF+T2Y*YDIF+T2Z*ZDIF
ETACK=ETA(2)-ETA(4)
GO TO 432
YES
IS
ETACK.NE.0.0
NO
XIO=0.0
GO TO 433
432
XIO=.333333333*(XI(4)*(ETA(1)-ETA(2))+XI(2)*(ETA(4)-ETA(1)))/(ETA(2)-ETA(4))
433
ETAO=-.333333333*ETA(1)
REPEAT TO 1020
FOR
J=1,1+1,....,4
XI(J)=XI(J)-XIO
1020
ETA(J)=ETA(J)-ETAO
XCENT=AVX+T1X*XIO+T2X*ETAO
YCENT=AVY+T1Y*XIO+T2Y*ETAO
ZCENT=AVZ+T1Z*XIO+T2Z*ETAO
XI3M1=XI(3)-XI(1)
ETA2M4=ETA(2)-ETA(4)
AREA=0.5*XI3M1*ETA2M4
AREAT=AREAT+AREA

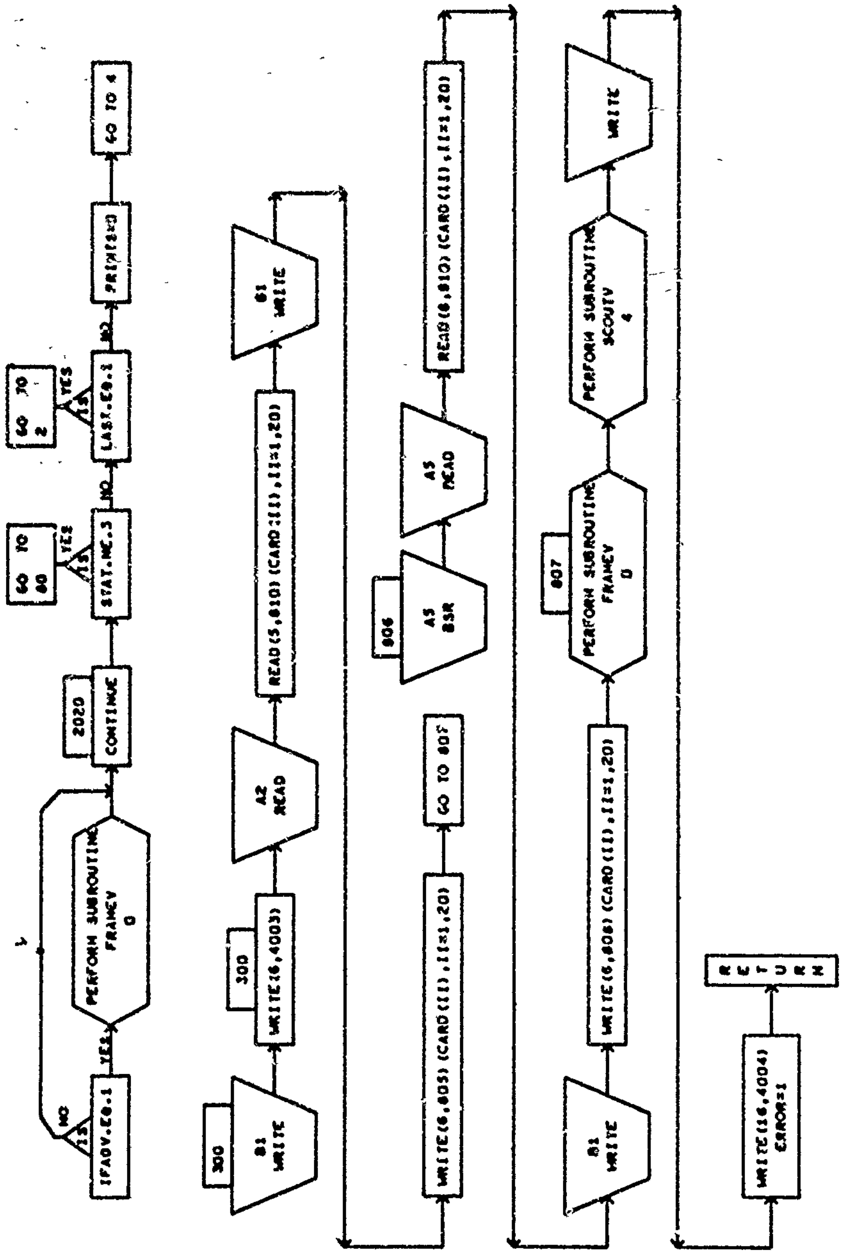

DELVOL=AREA*HY*YCENT
VOL=VOL+DELVOL
L=L+1
PRINTS.EQ.0
IS
YES
GO TO 1770
NO
1700
CONTINUE
NPRT.GE.9
GO TO 1750
NPRT=NPRT+1
L.EQ.1
GO TO 1760
B1
WRITE
WRITE(6,4005)I,XIN,NX,XCENT,AREA,L,YIN,NY,YCENT,DELVOL,ZIN,NZ,ZCENT,VOL
GO TO 1770
1750
NPRT=0
PERFORM SUBROUTINE HEADR2
WRITE(6,4002)
1760
WRITE(6,4010)N,I,XIN,NX,XCENT,AREA,L,YIN,NY,YCENT,DELVOL,ZIN,NZ,ZCENT,VOL
1770
AREA.LT.0.1E-09
GO TO 2000
IFADV.EQ.0
GO TO 471
NOSCAL.EQ.0
GO TO 505
PERFORM SUBROUTINE STOPTV
PERFORM SUBROUTINE GRITEV
PERFORM SUBROUTINE XSCALV XLG,XRG,24,8
PERFORM SUBROUTINE YSCALV YBS,YTG,0,24
GO TO 511
505
PERFORM SUBROUTINE GRIDIV 2,XLG,XRG,YBG,YTG,DXG,DYG,NG,NG,IG,JG,NXG,NYG
REPEAT TO 310 FOR I=1,I+2,...,3


510
PERFORM SUBROUTINE
APRNTV
5,-12,30,VTITLE,6,66
REPEAT TO 520
FOR
II=1,1+1,....,3
520
PERFORM SUBROUTINE
PRINTV
29,HTITLE,391,8
511
CONTINUE
GO TO
521
YES
IS
IFRAME.EQ.1.AND.ISTART.EQ.1
NO
A2
READ
READ(5,522)(HLABEL(II),II=1,15),TYPE
GO TO
306
YES
IS
TYPE.NE.37
NO
ISTART=1
521
REPEAT TO 523
FOR
II=1,1+1,....,3
PERFORM SUBROUTINE
PRINTV
-45,45HHYPERSONICARBITRARY-BODYAERODYNAMICPROGRAM,330,102
523
PERFORM SUBROUTINE
PRINTV
52,HLABEL,248,1007
GO TO
525
YES
IS
IFRAME.NE.1
NO
PERFORM SUBROUTINE
SCOUTV
4
WRITE
WRITE(16,524)XIN(1),XIN(4)
525
IFADV=0
471
NXO=THIRD(NX,NY,NZ,PSI,THETA,PHI)
GO TO
571
YES
IS
NXO.LE.0.0.AND.ISHAD.EQ.0
NO
530
YO1=FIRST(XIN(1),YIN(1),ZIN(1),A1,A2,A3)
YO2=FIRST(XIN(2),YIN(2),ZIN(2),A1,A2,A3)
YO3=FIRST(XIN(3),YIN(3),ZIN(3),A1,A2,A3)
YO4=FIRST(XIN(4),YIN(4),ZIN(4),A1,A2,A3)
ZO1=FIRST(XIN(1),YIN(1),ZIN(1),A4,A5,A6)
ZO2=FIRST(XIN(2),YIN(2),ZIN(2),A4,A5,A6)
ZO3=FIRST(XIN(3),YIN(3),ZIN(3),A4,A5,A6)
ZO4=FIRST(XIN(4),YIN(4),ZIN(4),A4,A5,A6)
YIN2(1)=YO1
YIN2(2)=YO2
YIN2(3)=YO3
YIN2(4)=YO4


ZINZ(1)=Z01
ZINZ(2)=Z02
ZINZ(3)=Z03
ZINZ(4)=Z04
PERFORM SUBROUTINE
APLOTV
,YINZ,ZINZ,1,1,1,MARKPT,IERR
GO TO
571
IS
YES
ICS.EQ.3
NO
GO TO
540
ICS.EQ.0.OR.ICS.EQ.1
GO TO 511
540
IX1=NXV(YO1)
IY1=NYV(ZO1)
PERFORM SUBROUTINE
SCERRV
KX,KY
IX2=NXV(YO2)
IY2=NYV(ZO2)
PERFORM SUBROUTINE
SCERRV
KX1,KY1
KKXY=KX*KY*KX1*KY1
GO TO
541
KKXY.NE.0
NXO.GT.0.0
PERFORM SUBROUTINE
LINEV
IX1,IY1,IX2,IY2
NXO.LE.0.0
PERFORM SUBROUTINE
DOTLNV
IX1,IY1,IX2,IY2
541
CONTINUE
GO TO
550
ICS.EQ.0.OR.ICS.EQ.2.OR.ICS.EQ.4
GO TO 551
550
IX1=NXV(YO2)
IY1=NYV(ZO2)
PERFORM SUBROUTINE
SCERRV
KX,KY
IX2=NXV(YO3)
IY2=NYV(ZO3)
PERFORM SUBROUTINE
SCERRV
KX1,KY1
KKXY=KX*KY*KX1*KY1
GO TO
551
KKXY.NE.0
NXO.GT.0.0
PERFORM SUBROUTINE
LINEV
IX1,IY1,IX2,IY2
NXO.LE.0.0
PERFORM SUBROUTINE
DOTLNV
IX1,IY1,IX2,IY2
551
CONTINUE


ICS.EQ.0.OR.ICS.EQ.1.OR.ICS.EQ.4
IS
YES
NO
GO TO 368
GO TO 361
368
IX1=NXV(YO3)
IY1=NYV(ZO3)
PERFORM SUBROUTINE SCERRV KX,KY
IX2=NXV(YO4)
IY2=NYV(ZO4)
PERFORM SUBROUTINE SCERRV KX1,KY1
KKXY=KX*KY+KX1*KY1
KKXY.NE.0
GO TO 361
NXO.GT.0.0
PERFORM SUBROUTINE LINEV IX1,IY1,IX2,IY2
NXO.LE.0.0
PERFORM SUBROUTINE DOTLNV IX1,IY1,IX2,IY2
361
CONTINUE
ICS.EQ.0.OR.ICS.EQ.2
GO TO 370
GO TO 371
370
IX1=NXV(YO1)
IY1=NYV(ZO1)
PERFORM SUBROUTINE SCERRV KX,KY
IX2=NXV(YO4)
IY2=NYV(ZO4)
PERFORM SUBROUTINE SCERRV KX1,KY1
KKXY=KX*KY+KX1*KY1
KKXY.NE.0
GO TO 371
NXO.GT.0.0
PERFORM SUBROUTINE LINEV IX1,IY1,IX2,IY2
NXO.LE.0.0
PERFORM SUBROUTINE DOTLNV IX1,IY1,IX2,IY2
371
CONTINUE


IREFL.EQ.0.OR.IRFLG.EQ.3
IS
YES
GO TO 2000
NO
IREFL.EQ.2.AND.IRFLG.EQ.1
GO TO 500
IREFL.EQ.2.AND.IRFLG.EQ.2
GO TO 602
REPEAT TO 580 FOR II=1,1+1,...,4
580
YIN(II)=-YIN(II) NY=-NY
GO TO 604
600
REPEAT TO 601 FOR II=1,1+1,...,4
YIN(II)=-YIN(II)
601
ZIN(II)=-ZIN(II) NY=-NY NZ=-NZ
GO TO 604
602
REPEAT TO 603 FOR II=1,1+1,...,4
603
YIN(II)=-YIN(II) NY=-NY
604
IRFLG=IRFLG+1
IREFL.EQ.1
IRFLG=3
GO TO 471
2000
CONTINUE
2001
CONTINUE
STAT.LT.2
GO TO 480
NPRT=NPRT+1
81
WRITE
WRITE(6,472)AREA,T,L,VOL NN=NN-1 NX=1
IAREA.EQ.0
GO TO 475
PERFORM SUBROUTINE SCOUTV 3
WRITE
WRITE(16,472)AREA,T,L
475
CONTINUE
IFRAME.EQ.2
IFADV=1
480
CONTINUE
IFRAME.EQ.1
IFADV=1
485
CONTINUE


IFADV.EQ.1
IS
NO
YES
PERFORM SUBROUTINE FRAMEV 0
2020
CONTINUE
STAT.NE.3
IS
YES
GO TO 60
NO
LAST.EQ.1
IS
YES
GO TO 2
NO
PRINT=0
GO TO 4
300
B1 WRITE
300
WRITE(6,4003)
A2 READ
READ(5,810)(CARD(II),II=1,20)
B1 WRITE
WRITE(6,805)(CARD(II),II=1,20)
GO TO 807
806
A5 BSR
A5 READ
READ(8,810)(CARD(II),II=1,20)
B1 WRITE
WRITE(6,808)(CARD(II),II=1,20)
807
PERFORM SUBROUTINE FRAMEV 0
PERFORM SUBROUTINE SCOUTV 4
WRITE
WRITE(16,4004) ERROR=1
RETURN

SYMBOLS USED IN SUBROUTINE PICTUR

AFLAG	L	U	IMPUT DATA READ CONTROL FLAG	PICTUR
AREA	R	U	ELEMENT AREA	PICTUR
AREAT	R	U	TOTAL AREA	PICTUR
AVX	R	U	AVERAGE POINT COORDINATE-X	PICTUR
AVY	R	U	AVERAGE POINT COORDINATE-Y	PICTUR
AVZ	R	U	AVERAGE POINT COORDINATE-Z	PICTUR
A1	R	U	ROTATION MATRIX CONSTANT	PICTUR
A2	R	U	ROTATION MATRIX CONSTANT	PICTUR
A3	R	U	ROTATION MATRIX CONSTANT	PICTUR
A4	R	U	ROTATION MATRIX CONSTANT	PICTUR
A5	R	U	ROTATION MATRIX CONSTANT	PICTUR
A6	R	U	ROTATION MATRIX CONSTANT	PICTUR
A7	R	U	ROTATION MATRIX CONSTANT	PICTUR
A8	R	U	ROTATION MATRIX CONSTANT	PICTUR
A9	R	U	ROTATION MATRIX CONSTANT	PICTUR
BFLAG	L	U	IMPUT DATA READ CONTROL FLAG	PICTUR
CARD	R	D	ARRAY FOR READING IN 80 COLUMN CARD	PICTUR
CASE	I	C	CASE NUMBER	PICTUR
COSPHI	R	U	COSINE OF PHI	PICTUR
COSPSI	R	U	COSINE OF PSI	PICTUR
COSTH	R	U	COSINE OF THETA	PICTUR
D	R	U	CORNER POINT PROJECTION DISTANCE	PICTUR
DELVOL	R	U	ELEMENT VOLUME CONTRIBUTION	PICTUR
DELX	R	U	GEOMETRY DATA X-INCREMENT	PICTUR
DELY	R	U	GEOMETRY DATA Y-INCREMENT	PICTUR
DELZ	R	U	GEOMETRY DATA Z-INCREMENT	PICTUR
DXG	R	U	GRID DELTA-X INCREMENT	PICTUR
DYG	R	U	GRID DELTA-Y INCREMENT	PICTUR
ERROR	I	C	ERROR FLAG	PICTUR
ETA	R	D	COORDINATE IN ELEMENT COORDINATE SYSTEM	PICTUR
ETACK	R	U	ETA CHECK PARAMETER	PICTUR
ETAO	R	U	COORDINATE IN ELEMENT COORDINATE SYSTEM	PICTUR
ETA2M4	R	U	CONSTANT IN AREA EQUATION	PICTUR
HLABEL	R	D	HORIZONTAL LABEL	PICTUR
HTITLE	R	D	VERTICAL LABEL	PICTUR
I	I	U	ELEMENT NUMBER IN COLUMN	PICTUR

SYMBOLS USED IN SUBROUTINE PICTUR

IA	I	U	FLAG TO CONTROL SHIFTING OF COLUMN DATA POINTS FOR IORIEN=3	PICTUR
IAREA	I	U	SURFACE AREA PRINT FLAG	PICTUR
IB	I	U	FLAG TO CONTROL SHIFTING OF COLUMN DATA POINTS FOR IORIEN=2	PICTUR
ICS	I	U	POINT CORRECT FLAG	PICTUR
IDUM	I	U	DUMMY VARIABLE	PICTUR
IERR	I	U	ERROR FLAG	PICTUR
IFACT	I	U	SCALE FACTOR FLAG	PICTUR
IFADV	I	U	FRAME FLAG	PICTUR
IFRAME	I	U	FRAME ADVANCE FLAG	PICTUR
IG	I	U	VERTICAL GRID LINE LABEL CONTROL FLAG	PICTUR
IIA	I	U	DATA SHIFTING CONTROL PARAMETER (IORIEN=3)	PICTUR
IIB	I	U	DATA SHIFTING CONTROL PARAMETER (IORIEN=2)	PICTUR
IORIEN	I	U	ELEMENT ORIENTATION (NOT USED)	PICTUR
IPIC	I	U	FRAME NUMBER	PICTUR
IQUAD	I	U	QUADRILATERAL PLOT FLAG	PICTUR
IREFL	I	U	REFLECTION ELEMENTS CONTROL FLAG	PICTUR
IREW8	I	U	TAPE 8 REWIND FLAG	PICTUR
IRFLG	I	U	REFLECTION CONTROL FLAG	PICTUR
ISHAD	I	U	SHADOW ELEMENT FLAG	PICTUR
ISTART	I	U	CONTROL FLAG	PICTUR
ISTAT3	I	U	NUMBER OF STATUS = 3 POINTS IN DECK	PICTUR
ITAPE	I	U	GEOMETRY SOURCE FLAG	PICTUR
IX1	I	U	X-RASTER COORDINATE OF FIRST POINT	PICTUR
IX2	I	U	X-RASTER COORDINATE OF SECOND POINT	PICTUR
IY1	I	U	Y-RASTER COORDINATE OF FIRST POINT	PICTUR
IY2	I	U	Y-RASTER COORDINATE OF SECOND POINT	PICTUR
JG	I	U	HORIZONTAL GRID LINE LABEL CONTROL FLAG	PICTUR
KKXY	I	U	OFF-SCALE DETECTION FLAG	PICTUR
KLCT	I	U	COUNTER	PICTUR
KX	I	U	OFF-SCALE DETECTION FLAG	PICTUR
KX1	I	U	OFF-SCALE DETECTION FLAG	PICTUR
KY	I	U	OFF-SCALE DETECTION FLAG	PICTUR
KY1	I	U	OFF-SCALE DETECTION FLAG	PICTUR
L	I	U	ELEMENT NUMBER	PICTUR
LAST	I	U	LAST PLOT CONTROL FLAG	PICTUR
M	I	U	DATA READ IN CONTROL FLAG	PICTUR

SYMBOLS USED IN SUBROUTINE PICTUR

MARKPT	I	U	PLOTTING SYMBOL CODE	PICTUR
MC	I	U	DATA READ IN CONTROL NUMBER	PICTUR
MG	I	U	HORZONTAL LINE EMPHASIZE FLAG	PICTUR
MMIN	I	U	NUMBER OF ELEMENTS IN A COLUMN	PICTUR
N	I	U	COLUMN NUMBER	PICTUR
NCAM	I	U	CAMERA SELECTION FLAG	PICTUR
NG	I	U	VERTICAL LINE EMPHASIZE FLAG	PICTUR
NN	I	U	COLUMN ELEMENT COUNTER	PICTUR
NN2	I	U	COUNTER	PICTUR
NOSCAL	I	U	NO GRID FLAG	PICTUR
NPRT	I	U	LINE COUNTER	PICTUR
NX	R	U	ELEMENT DIRECTION COSINE-X	PICTUR
NXG	I	U	NUMBER OF CHARACTERS IN X-SCALE NUMBER LABELS	PICTUR
NXO	R	U	DIRECTION COSINE OUT OF PLANE OF PAPER	PICTUR
NY	R	U	ELEMENT DIRECTION COSINE-Y	PICTUR
NYG	I	U	NUMBER OF CHARACTERS IN Y-SCALE NUMBER LABELS	PICTUR
NZ	R	U	ELEMENT DIRECTION COSINE-Z	PICTUR
PAGE	I	C	PAGE NUMBER	PICTUR
PD	R	U	CORNER POINT PROJECTION DISTANCE	PICTUR
PHI	R	U	ROLL ANGLE, DEGREES	PICTUR
PRINTS	I	U	ELEMENT DATA PRINT FLAG	PICTUR
PSI	R	U	YAW ANGLE	PICTUR
RFLAG	L	U	INPUT DATA READ CONTROL FLAG	PICTUR
SINPHI	R	U	SIN OF PHI	PICTUR
SINPSI	R	U	SIN OF PSI	PICTUR
SINTH	R	U	SIN OF THETA	PICTUR
STAT	I	U	COORDINATE POINT STATUS FLAG	PICTUR
STATT	I	U	COORDINATE POINT STATUS FLAG	PICTUR
SYMFCT	I	U	SYMMETRY FLAG	PICTUR
T	R	U	UNIT VECTOR	PICTUR
THETA	R	U	PITCH ANGLE	PICTUR
TITLE	R	C	TITLE	PICTUR
TYPE	I	U	CARD TYPE NUMBER	PICTUR
T1X	R	U	X-COMPONENT OF VECTOR T1	PICTUR
T1Y	R	U	Y-COMPONENT OF VECTOR T1	PICTUR
T1Z	R	U	Z-COMPONENT OF VECTOR T1	PICTUR

SYMBOLS USED IN SUBROUTINE PICTUR

T2X	R	U	X-COMPONENT OF VECTOR T2	PICTUR
T2Y	R	U	Y-COMPONENT OF VECTOR T2	PICTUR
T2Z	R	U	Z-COMPONENT OF VECTOR T2	PICTUR
VN	R	U	VECTOR LENGTH	PICTUR
VOL	R	U	TOTAL VOLUME	PICTUR
VTITLE	R	D	VERTICAL SCALE TITLE	PICTUR
X	R	U	X-COORDINATE	PICTUR
XA	R	D	X-COORDINATE	PICTUR
XB	R	D	X-COORDINATE	PICTUR
XCENT	R	U	ELEMENT CENTROID COORDINATE-X	PICTUR
XDIF	R	U	COORDINATE DIFFERENCE-X	PICTUR
XI	R	D	COORDINATE IN ELEMENT COORDINATE SYSTEM	PICTUR
XIN	R	D	ELEMENT COORDINATES-X	PICTUR
XIO	R	U	CENTROID IN ELEMENT COORDINATE SYSTEM	PICTUR
XI3M1	R	U	CONSTANT FOR AREA EQUATION	PICTUR
XLG	R	U	VALUE OF LEFT SIDE OF HORIZONTAL SCALE	PICTUR
XPA	R	U	COORDINATE OF ELEMENT CORNER POINT	PICTUR
XRG	R	U	VALUE OF RIGHT SIDE OF HORIZONTAL SCALE	PICTUR
XSC	R	U	X SCALE FACTOR	PICTUR
XX	R	U	X-COORDINATE	PICTUR
Y	R	U	Y-COORDINATE	PICTUR
YA	R	D	Y-COORDINATE	PICTUR
YB	R	D	Y-COORDINATE	PICTUR
YBG	R	U	VALUE OF BOTTOM OF VERTICAL SCALE	PICTUR
YCENT	R	U	ELEMENT CENTROID COORDINATE-Y	PICTUR
YDIF	R	U	COORDINATE DIFFERENCE-Y	PICTUR
YIN	R	D	ELEMENT COORDINATES-Y	PICTUR
YIN2	R	D	Y-COORDINATE FOR PLOT	PICTUR
YO1	R	U	Y-COORDINATE FOR PLOT-POINT 1	PICTUR
YO2	R	U	Y-COORDINATE FOR PLOT-POINT 2	PICTUR
YO3	R	U	Y-COORDINATE FOR PLOT-POINT 3	PICTUR
YO4	R	U	Y-COORDINATE FOR PLOT-POINT 4	PICTUR
YPA	R	U	COORDINATE OF ELEMENT CORNER POINT	PICTUR
YSC	R	U	Y-SCALE FACTOR	PICTUR
YTG	R	U	VALUE OF TOP OF VERTICAL SCALE	PICTUR
YY	R	U	Y-COORDINATE	PICTUR

SYMBOLS USED IN SUBROUTINE PICTUR

Z	R	U	Z-COORDINATE	PICTUR
ZA	R	D	Z-COORDINATE	PICTUR
ZB	R	D	Z-COORDINATE	PICTUR
ZCENT	R	U	ELEMENT CENTROID COORDINATE-Z	PICTUR
ZDIF	R	U	COORDINATE DIFFERENCE-Z	PICTUR
ZIN	R	D	ELEMENT COORDINATES-Z	PICTUR
ZIN2	R	D	Z-COORDINATE FOR PLOT	PICTUR
Z01	R	U	Z-COORDINATE FOR PLOT-POINT 1	PICTUR
Z02	R	U	Z-COORDINATE FOR PLOT-POINT 2	PICTUR
Z03	R	U	Z-COORDINATE FOR PLOT-POINT 3	PICTUR
Z04	R	U	Z-COORDINATE FOR PLOT-POINT 4	PICTUR
ZPA	R	U	COORDINATE OF ELEMENT CORNER POINT	PICTUR
ZSC	R	U	Z-SCALE FACTOR	PICTUR
ZZ	R	U	Z-COORDINATE	PICTUR

30. SUBROUTINE HEADR2 (DECK GRPC)

a. Algorithm

This routine provides the title at the top of each page of the output and advances the page counter. This routine is very similar to the HEADER routine.

b. Input/Output

Program header is printed at top of page on output Tape 6.

c. Error

None

d. Subroutines Required

None

e. Argument List

None

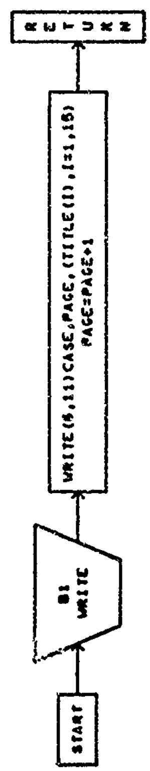
f. Length

350 bytes

DECK GRPC

```
      SUBROUTINE HEADR2
C
      DIMENSION  TITLE(15)
C
      COMMON CASE,TITLE,PAGE
C
      INTEGER  PAGE, CASE
C
C  PRINT OUT HEADER AT TOP OF EACH PAGE OF OUTPUT
      WRITE (6,11) CASE,PAGE,(TITLE(I),I=1,15)
   11 FORMAT (1H1,5X,39HQUADRILATERAL CHARACTERISTICS - PICTURE
     1 16H DRAWING PROGRAM ,/1H0,5X,6H  CASE,I5,80X,5HPAGE I4,/
     2 1H0,14A4,A3)
C
C  STEP PAGE NUMBER BY ONE
      PAGE = PAGE + 1
C
      RETURN
      END
```

SUBROUTINE HEADR2



```
SYMBOLS USED IN SUBROUTINE HEADR2                                  

CASE     I  C  CASE NUMBER                                          HEADR2
PAGE     I  C  PAGE NUMBER                                          HEADR2
TITLE    R  C  TITLE                                                HEADR2
```

31. SUBROUTINE PLOT (DECK GRPD)

This routine is used to produce graphically plotted data as obtained from the aerodynamics part of the program.

a. Algorithm

Read in plotter control cards. As directed, read aerodynamic data from Tape 9. Prepare plot scales and grids. Plot data and connect data points as directed.

b. Input/Output

Data Source Control Card (Type 41), Vertical-Title Card (Type 44), Horizontal-Title Card (Type 45), Plotting-Grid Data Card (Type 45), Plot Control Array Card (Type 47), and Horizontal-Lable Card(s) Type 48).
Output plots are on the SC-4020 tape.

c. Error

An error condition occurs when the card type number is wrong.

d. Subroutines Required

None

e. Argument List

None

f. Length

10772 bytes

DECK GRPD

```
      SUBROUTINE PLOT
C      POINT PLOTTER PROGRAM. UP TO 12 ARRAYS CAN BE INPUT
C      AND ANY ONE OF THEM PLOTTED AGAINST ANY OTHER
C
      DIMENSION A(20),B(20),C(20),D(20),E(20),F(20),AA(20),BB(20),
     1 CC(20),DD(20),EE(20),FF(20),X(20),Y(20),I(100),TITLE(900),W(10),
     3 PRINT(18),TITLE2(15)
      COMMON  CASE,TITLE2,PAGE,ERROR
      INTEGER  ERROR,TYPE,PAGE,CASE
C
      CALL CAMRAV (9)
      REWIND 1
      REWIND 10
      REWIND 9
C     READ FIRST CONTROL CARD
    1 READ (5,2) NC,IT
    2  FORMAT (I4,66XI2)
C     CHECK TYPE OF CARD
      IF (IT .EQ. 41) GO TO 5
C
C     TYPE CARD GOOFY. PRINT ERROR HEADER AND LEAVE
      IT = 41
    3 CALL CAMRAV(9)
      CALL FRAMEV (0)
      CALL SCOUTV
      CALL SCSETV (4)
      WRITE (16,4) IT,IT
    4 FORMAT (1H4,15X30HFOR SOME ODD REASON, TYPE CARDI3,1X
     1  15HDOES NOT HAVE AI3,31H IN COLUMN 71-72. BETTER LUCK N
     2  9HEXT TIME. )
      WRITE (6,48) IT,IT
   48 FORMAT (1H0,15X30HFOR SOME ODD REASON, TYPE CARDI3,1X
     1  15HDOES NOT HAVE AI3,31H IN COLUMN 71-72. BETTER LUCK N
     2  9HEXT TIME. )
      READ (5,49) (TITLE(J),J=1,20)
```

DECK GRPD

```
   49 FORMAT (20A4)                                               GRPD 0360
      WRITE (6,50) (TITLE(J),J=1,20)                              GRPD 0370
   50 FORMAT (1HO,26X36HWRITTEN BELOW IS THE IMAGE OF THE CA      GRPD 0380
     1  30HRD FOLLOWING THE INCORRECT ONE//20X,20A4)              GRPD 0390
      ERROR = 1                                                   GRPD 0400
      GO TO 101                                                   GRPD 0410
C                                                                 GRPD 0420
C     CHECK FOR INPUT TAPE, IF ANY                                GRPD 0430
    5 I(20) = 0                                                   GRPD 0440
      ICNT  = 0                                                   GRPD 0450
      IF (NC) 28,33,7                                             GRPD 0460
   33 READ(10)NC,NC,NC,NC,IT                                      GRPD 0470
      WRITE (6,44) NC                                             GRPD 0480
   44 FORMAT (1HO,31X34HTAPE 10 JUST INSTRUCTED ME TO READI4      GRPD 0490
     1  ,18H CARDS FROM TAPE 9 )                                  GRPD 0500
C                                                                 GRPD 0510
C     CHECK TYPE                                                  GRPD 0520
      IF (IT .EQ. 42) GO TO 43                                    GRPD 0530
C                                                                 GRPD 0540
C     TYPE ERROR                                                  GRPD 0550
      IT = 42                                                     GRPD 0560
      GO TO 3                                                     GRPD 0570
   43 DO 34 J = 1,NC                                              GRPD 0580
   34  READ (9) A(J),B(J),C(J),D(J),E(J),F(J),AA(J),BB(J),         GRPD 0590
     1  CC(J),DD(J),EE(J),FF(J),IT                                GRPD 0600
C                                                                 GRPD 0610
C      CHECK TYPE (IT) OF CARD JUST READ                          GRPD 0620
      IF (IT .EQ. 43) GO TO 71                                    GRPD 0630
C                                                                 GRPD 0640
C      TYPE ERROR                                                 GRPD 0650
      IT = 43                                                     GRPD 0660
       GO TO 3                                                    GRPD 0670
   71  CONTINUE                                                   GRPD 0680
       WRITE (6,45) NC                                            GRPD 0690
   45  FORMAT (1HO,42X11HI JUST READI4,18H CARDS FROM TAPE 9)     GRPD 0700
      IF (I(20) .NE. 0) GO TO 78                                  GRPD 0710
```

PLOT

DECK ARDX

```
      READ (5,103) AR , LAMBDA , M , SWBYS , GAMMAT , S , K , TYPE
      IF (TYPE .NE. 17) GO TO 1000
      READ (5,104) CR , R , BETA , CLALS , UPWASH , XBTBYC , TYPE
      IF (TYPE .NE. 18) GO TO 1000
      READ (5,106) KBW , Q, TYPE
  106 FORMAT (2F10.0, 50X, I2)
      IF (TYPE .NE. 19) GO TO 1000
  107 CONTINUE
C
C ****************************************************************
C ***** WING OR TAIL CONTRIBUTION
C       THE SLENDER BODY THEORY IS USED TO DETERMINE KWB.  THE EXPRESSION
C       IS EQ. (14) OF NACA REPORT 1307.
C
      D = 2.0 * R
      RS = R / S
      SR = S / R
      KWB = 2.0 / PI * (    (1.0 + RS ** 4) * (0.5 * ATAN (0.5 * (SR - RS
     A      )) + PI / 4.0) - RS ** 2 * (SR - RS + 2.0 * ATAN (RS))    )
     B      / (1.0 - RS) ** 2
C
C ****CALCULATE KBW ***********************************************
C       SEVERAL EXPRESSIONS ARE TAKEN FROM NACA REPORT 1307 FOR KBW.
C       THIS ROUTINE GIVES THE FOLLOWING OPTIONS AS A FUNCTION OF THE
C       INTEGER ITYPE (IF EQ. (22) OF REPORT 1307 IS NOT SATISFIED,
C       SLENDER BODY THEORY IS AUTOMATICALLY USED UNLESS KBW IS INPUT
C       BY THE USER).
C
C ITYPE  REFERENCE FOR KBW
C   0      USER LOADS A VALUE OF KBW
C   1      NACA REPT. 1307 EQ. (21)  (SLENDER BODY THEORY)
C   2      NACA REPT. 1307 EQ. (24)    SUPERSONIC LEADING EDGE
C                              (26)    SUBSONIC LEADING EDGE
C              HALF-PLANFORM IS A TRAPEZOID.  WING/TAIL ON LONG BODY.
C   3      NACA REPT. 1307 EQ. (27)
C              RECTANGULAR PLANFORM.  WING/TAIL ON LONG BODY.
```

DECK AROX

```
C   4      NACA REPT. 1307 EQ. (28)    SUBSONIC LEADING EDGE               AROX  1080
C                          EQ. (29)    SUPERSONIC LEADING EDGE             AROX  1090
C                TRIANGULAR PLANFORM.  WING/TAIL ON LONG BODY.             AROX  1100
C   5      NACA REPT. 1307 EQ. (30)    SUPERSONIC LEADING EDGE             AROX  1110
C                          EQ. (31)    SUBSONIC LEADING EDGE               AROX  1120
C                HALF-PLANFORM IS A TRAPEZOID.  NO AFTER-BODY FOR WING.    AROX  1130
      ITYPE = K                                                            AROX  1140
C                                                                          AROX  1150
C     TEST ITYPE FOR VALID RANGE                                           AROX  1160
C                                                                          AROX  1170
      IF (ITYPE .GE. 0 .AND. ITYPE .LE. 5) GO TO 72                        AROX  1180
      WRITE (6,71) ITYPE                                                   AROX  1190
   71 FORMAT(1H ,107H**** SUBROUTINE PLUNGE - THE FLAG ITYPE (WHICH CONT   AROX  1200
     2ROLS EQUATION USED TO CALCULATE KBW) IS INCORRECT AND =  ,I7 //)     AROX  1210
      GO TO 1000                                                           AROX  1220
   72 CONTINUE                                                             AROX  1230
C                                                                          AROX  1240
C                                                                          AROX  1250
C     TEST FOR EQ. (22) FOLLOWS.  IF TEST FAILS, SLENDER-BODY THEORY       AROX  1260
C     WILL BE USED AUTOMATICALLY.                                          AROX  1270
C                                                                          AROX  1280
      IF (ITYPE .GT. 1 .AND. (BETA * AR * (1.0 + LAMBDA) * (1.0 /          AROX  1290
     A    (BETA * M) + 1.0)) .LT. 4.0) ITYPE = 1                           AROX  1300
      I = ITYPE + 1                                                        AROX  1310
      GO TO (30, 2, 3, 5, 6, 8) , I                                        AROX  1320
C                                                                          AROX  1330
C     EQ. (21)                                                             AROX  1340
C                                                                          AROX  1350
    2 KBW = (    (1.0 - RS ** 2) ** 2 - 2.0 / PI * ( (1.0 + RS ** 4) *     AROX  1360
     A      (0.5 * ATAN (0.5 * (SR - RS)) + PI / 4.0) - RS ** 2 *          AROX  1370
     B      (SR - RS + 2.0 * ATAN (RS)) )     ) / (1.0 - RS) ** 2          AROX  1380
      GO TO 30                                                             AROX  1390
C                                                                          AROX  1400
C     EQ. (24)                                                             AROX  1410
C                                                                          AROX  1420
    3 BMI = BETA * M                                                       AROX  1430
```

DECK AROX

```
      IF (BM1 .LT. 1.0) GO TO 4
      BM2 = BM1 ** 2
      BMR2 = SQRT (BM2 - 1.0)
      T1 = (   1.0 + (1.0 + BM1) * BETA * D / CR   )
     A     / (   BM1 + (BM1 + 1.0) * BETA * D / CR   )
      T2 = 1.0 / BM1
      T1 = ARCOS (T1)
      T2 = ARCOS (T2)
      KBW = 8.0 * BM1 / (PI * BMR2 * (1.0 + LAMBDA) * BETA * D / CR *
     A       (SR - 1.0) * BETA * CLALW)
     B       * (    (BM1 / (1.0 + BM1)) * (((BM1 + 1.0) * BETA * D / CR
     C       + BM1) / BM1) ** 2 * T1
     D       + BMR2 / (BM1 + 1.0) * (SQRT(1.0 + 2.0 * BETA * D / CR)
     E       - 1.0)
     F       - BMR2 / BM1 * (BETA * D / CR) ** 2 * ACOSH(1.0 + CR/BETA/D)
     G       - BM1 / (1.0 + BM1) * T2   )
      GO TO 30
C
C     EQ. (26)
C
    4 BM = BM1
      T1 = SQRT ( (BM + (1.0 + BM) * BETA * D / CR) / BM )
      KBW = 16.0 * (BM / (1.0 + BM)) ** 2
     A       / (PI * (1.0 + LAMBDA) * BETA * D / CR * (SR - 1.0) * BETA
     B       * CLALW)
     C       * (    T1 ** 3 + T1 - 2.0 - (((1.0 + BM) * BETA * C / CR)
     D       / BM) ** 2 * ATANH (1.0 / T1)   )
      GO TO 30
C
C     EQ. (27)
C
    5 BA = BETA * AR
      BARS = BA * RS
      T1 = BA / (BA + SR - 1.0)
      T1 = ARCOS (T1)
      KBW = 2.0 / (PI * (CA - 0.5)) *
```

```
DECK AROX
      A       (    0.5 * (1.0 + BARS / (1.0 - RS)) ** 2 * T1                  AROX  1800
      B       - 0.5 * (BARS / (1.0 - RS)) ** 2 * ACOSH (1.0 + (1.0 - RS)     AROX  1810
      C       / BARS) - 0.5 - PI / 4.0 + 0.5 * SQRT (1.0 + 2.0 * BARS        AROX  1820
      D       / (1.0 - RS))    )                                              AROX  1830
       GO TO 30                                                               AROX  1840
C                                                                             AROX  1850
C      EQ. (28)                                                               AROX  1860
C                                                                             AROX  1870
     6 BA = BETA * AR                                                         AROX  1880
       BA4 = BA / 4.0                                                         AROX  1890
       IF (BA4 .GE. 1.0) GO TO 7                                              AROX  1900
       T1 = SQRT (1.0 - BA4 ** 2)                                             AROX  1910
       CALL ELP1 (T1, DUMMY, T1, NER1)                                        AROX  1920
C      ERROR TEST 3                                                           AROX  1930
       IF (NER1.NE.0) WRITE (6,22)                                            AROX  1940
   22  FORMAT (1H ,46H**** ELLIPTICAL INTEGRAL ERROR. T1 FROM PLUNGE          AROX  1950
      1  55H IS NOT LESS THAN ONE AND GREATER THAN OR EQUAL TO ZERO )         AROX  1960
       T2 = (BA4 / (BA4 + 1.0)) ** 2 / 2.0                                    AROX  1970
       T3 = SQRT (1.0 + 2.0 * (1.0 + BA4) * RS / (1.0 - RS))                  AROX  1980
       KBW = 8.0 * T1 / (PI * BA4) ** 2 *                                     AROX  1990
      A      (    T2 * T3 ** 3 - (BA / (BA + 4.0)) ** 2 + T2 * T3             AROX  2000
      B       - 2.0 * (BA4 * RS / (1.0 - RS)) ** 2 * ATANH (1.0 / T3)     )   AROX  2010
       GO TO 30                                                               AROX  2020
C                                                                             AROX  2030
C      EQ. (29)                                                               AROX  2040
C                                                                             AROX  2050
     7 T1 = 2.0 * (1.0 + BA4) * RS / (1.0 - RS)                               AROX  2060
       T2 = (1.0 + BA4 * T1) / (BA4 + BA4 * T1)                               AROX  2070
       T3 = 1.0 / BA4                                                         AROX  2080
       T2 = ARCOS (T2)                                                        AROX  2090
       T3 = ARCOS (T3)                                                        AROX  2100
       KBW = 1.0 / (PI * SQRT (BA4 ** 2 - 1.0)) *                             AROX  2110
      A       (    (BA / (BA + 4.0)) * (1.0 + T1) ** 2 * T2                   AROX  2120
      B       + SQRT ( (BA4 ** 2 - 1.0) * (1.0 + BA * RS / (1.0 - RS)) )      AROX  2130
      C       / (1.0 + BA4)  - BA4 * T3 / (1.0 + BA4)                         AROX  2140
      D       - SQRT (BA4 ** 2 - 1.0) * BA * (RS / (1.0 - RS)) ** 2           AROX  2150
```

DECK AROX

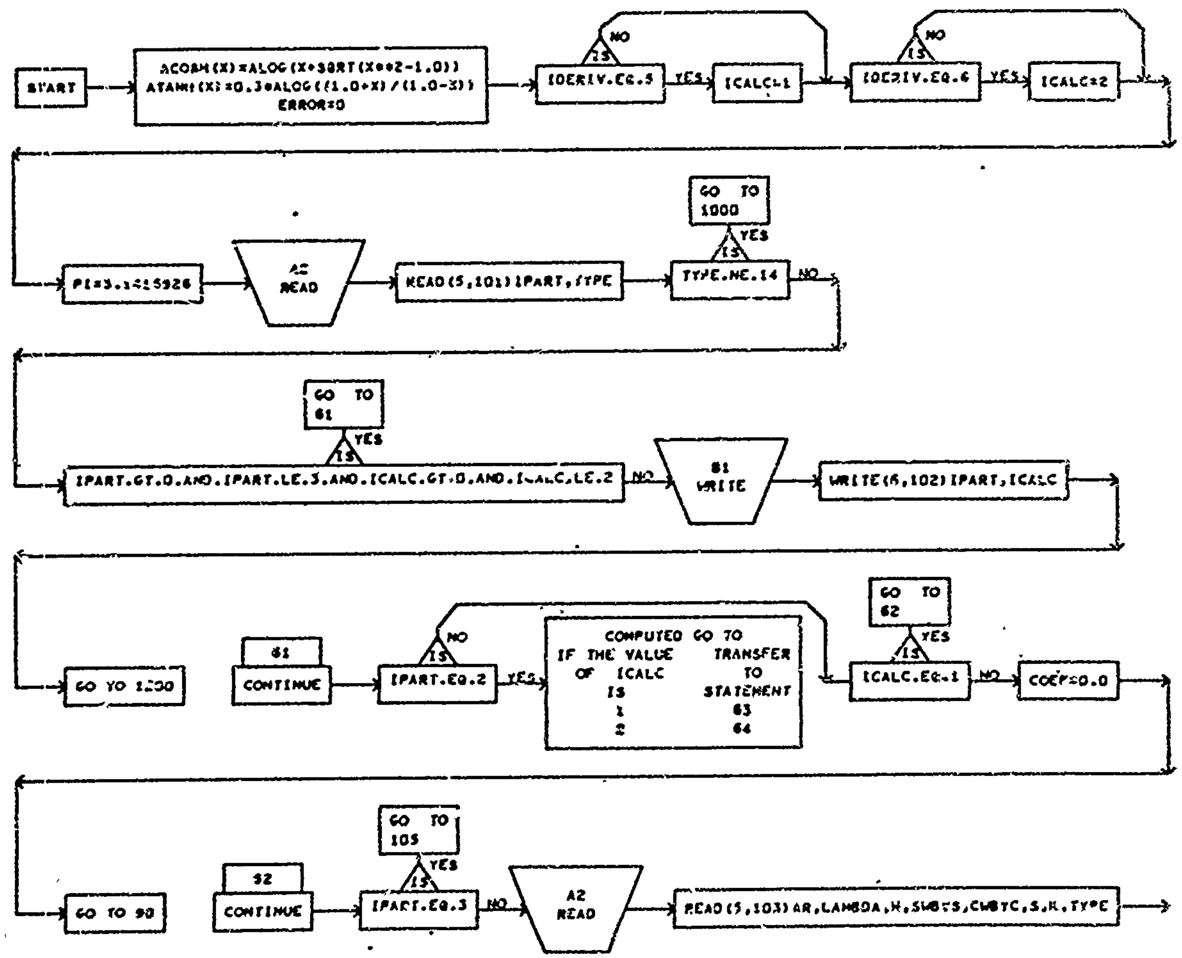
```
     E        * ACOSH (1.0 + 2.0 * (1.0 - RS) / BA / RS)
     F        - SQRT (BA4 ** 2 - 1.0) / (BA4 + 1.0)    )
      GO TO 30
C
C     EQNS. (30) AND (31)
C
    8 BDCR = BETA * D / CR
C
C     TEST BDCR.  IF BDCR GT 1.0, SET BDCR = 1.0 TO GET PROPER RESULT.
C     SEE PARAGRAPH FOLLOWING EQ. (31) OF NACA REPORT 1307.
C
      IF (BDCR .GT. 1.0) BDCR = 1.0
      BM    = BETA * M
      IF (BM .LT. 1.0) GO TO 9
C
C     EQ. (30)
C
      T1 = (BM + 1.0 / BDCR) / (1.0 + BM / BDCR)
      T2 = 1.0 / BM
      T3 = BDCR
      T4 = SQRT (BM ** 2 - 1.0)
      T1 = ARCOS (T1)
      T2 = ARCOS (T2
      T3 = ARSIN (T3)
      KBW = 8.0 * BDCR / (PI * T4 * BETA * CLALW * (1.0 + LAMBDA) *
     A        (SR - 1.0)) *
     B        (    (1.0 + BM / BDCR ) ** 2 * T1
     C        - (BM / BDCR) ** 2 * T2
     D        + BM / BDCR ** 2 * T4 * T3
     E        - T4 * ACOSH (1.0 / BDCR)
      GO TO 30
    9 CRBD = 1.0 / BDCR
      CRBD2 = CRBD ** 2
C
C     EQ. (31)
C
```

```
DECK AROX

      KBW = 16.0 * SQRT (BM) * BDCR / (PI * (BM + 1.0) * BETA * CLALW
     A       * (1.0 + LAMBDA) * (SR - 1.0)) *
     B       (    (1.0 + M * CR / D) * SQRT ((CRBD - 1.0)*(M*CR/D + 1.0))
     C       - CRBD2 * BM ** 1.5
     D       + BM * CRBD2 * (BM + 1.0) * (ATAN (SQRT(1.0 / BM))
     E       - ATAN (SQRT((CRBD - 1.0) / (M * CR / D + 1.0))))
     F       - (BM + 1.0) / SQRT (BM) * ATANH (SQRT(BM * (CRBD - 1.0) /
     G         (M * CR / D + 1.0)))    )
      GO TO 30
C
C     HOPEFULLY, BY THIS TIME, KBW HAS BEEN CALCULATED.
C
   30 CONTINUE
      IF (IPART .EQ. 3) GO TO 32
C
C***** CALCULATE WING CM ALPHA DOT ***************************************
      COEF = SWBYS * (KWB + KBW) * CWBYC ** 2 * CMWPR
      GO TO 90
C
C
C***** CALCULATE TAIL CONTRIBUTION TO C-SUB-M-SUB-BETA-DOTA
   32 CONTINUE
      COEF = 2.0 * Q * SWBYS * ( COS (0.017453292 * GAMMAT) ) ** 2
     A       * CLALW * (KWB + KBW) * XBTBYC ** 2 * UPWASH
      GO TO 90
C
C
C***** CONTRIBUTION OF BODY TO C-SUB-M-SUB-ALPHA-DOT   *****************
   63 CONTINUE
C
C     READ BODY DATA
C
      READ (5,104) VOLUME , SFRONT , LENGTH , XO , XC , C , TYPE
      IF (TYPE .NE. 20) GO TO 1000
      RATIO = (-2.0 * VOLUME / C / SFRONT) * (XO - XC) / LENGTH
     A       / (VOLUME / SFRONT / LENGTH - 1.0 + XO / LENGTH)
```

PLUNGE

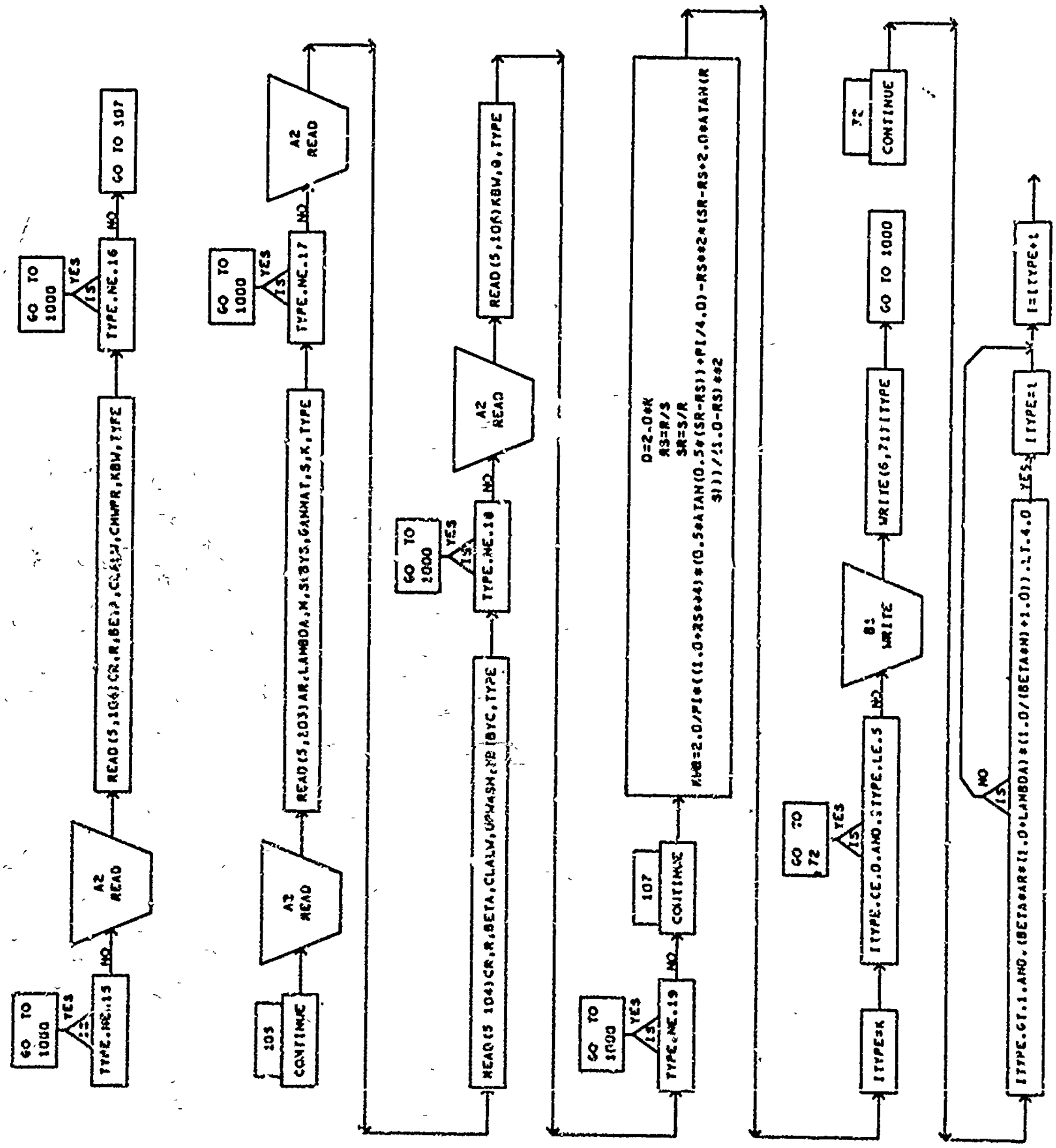
```
DECK AROX

      COEF = RATIO * CMA                                                 AROX  2880
      GO TO 90                                                           AROX  2890
C                                                                        AROX  2900
C***** BODY CY BETA DOT ************************************************* AROX  2910
   64 CONTINUE                                                           AROX  2920
C     THIS PART OF THE ROUTINE COMPUTES C-SUB-Y-SUB-BETA DOT             AROX  2930
C     THE BASIC REFERENCE IS NASA TMX-287.                               AROX  2940
C     THE REFERENCE SHOWS THAT WING AND TAIL CONTRIBUTIONS.CAN BE        AROX  2950
C     NEGLECTED, AND THAT THE FUSELAGE TERM CAN BE OBTAINED BY A         AROX  2960
C     SLENDER-BODY-THEORY RATIO MULTIPLIED BY C-SUB-Y-SUB-BETA.          AROX  2970
C                                                                        AROX  2980
C                                                                        AROX  2990
C     READ BODY DATA                                                     AROX  3000
C                                                                        AROX  3010
      READ (5,108) VOLUME , SFRONT , B , TYPE                            AROX  3020
  108 FORMAT (3F10.0 , 40X, I2)                                          AROX  3030
      IF (TYPE .NE. 21) GO TO 1000                                       AROX  3040
      COEF = (-2.0) * VOLUME / SFRONT * CYB / B                          AROX  3050
   90 IF (IDERIV .EQ. 5) CMADT = COEF                                    AROX  3060
      IF (IDERIV .EQ. 6) CYBDT = COEF                                    AROX  3070
      RETURN                                                             AROX  3080
 1000 ERROR = 1                                                          AROX  3090
      WRITE (6,1001)                                                     AROX  3100
 1001 FORMAT (1H , 39H**** SUBROUTINE PLUNGE SETS ERROR FLAG ///)        AROX  3110
      RETURN                                                             AROX  3120
      END                                                                AROX  3130
```

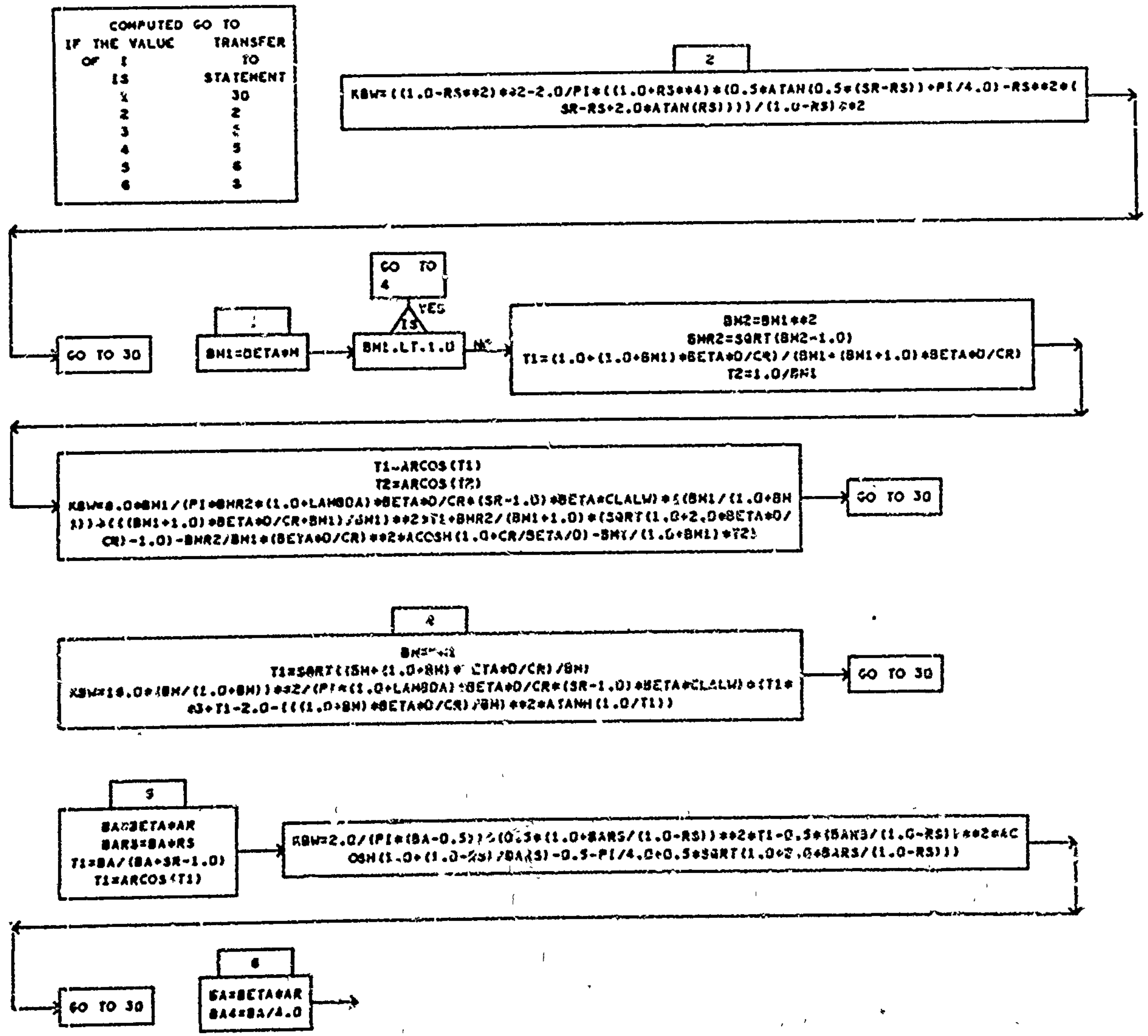
SUBROUTINE PLUNGE



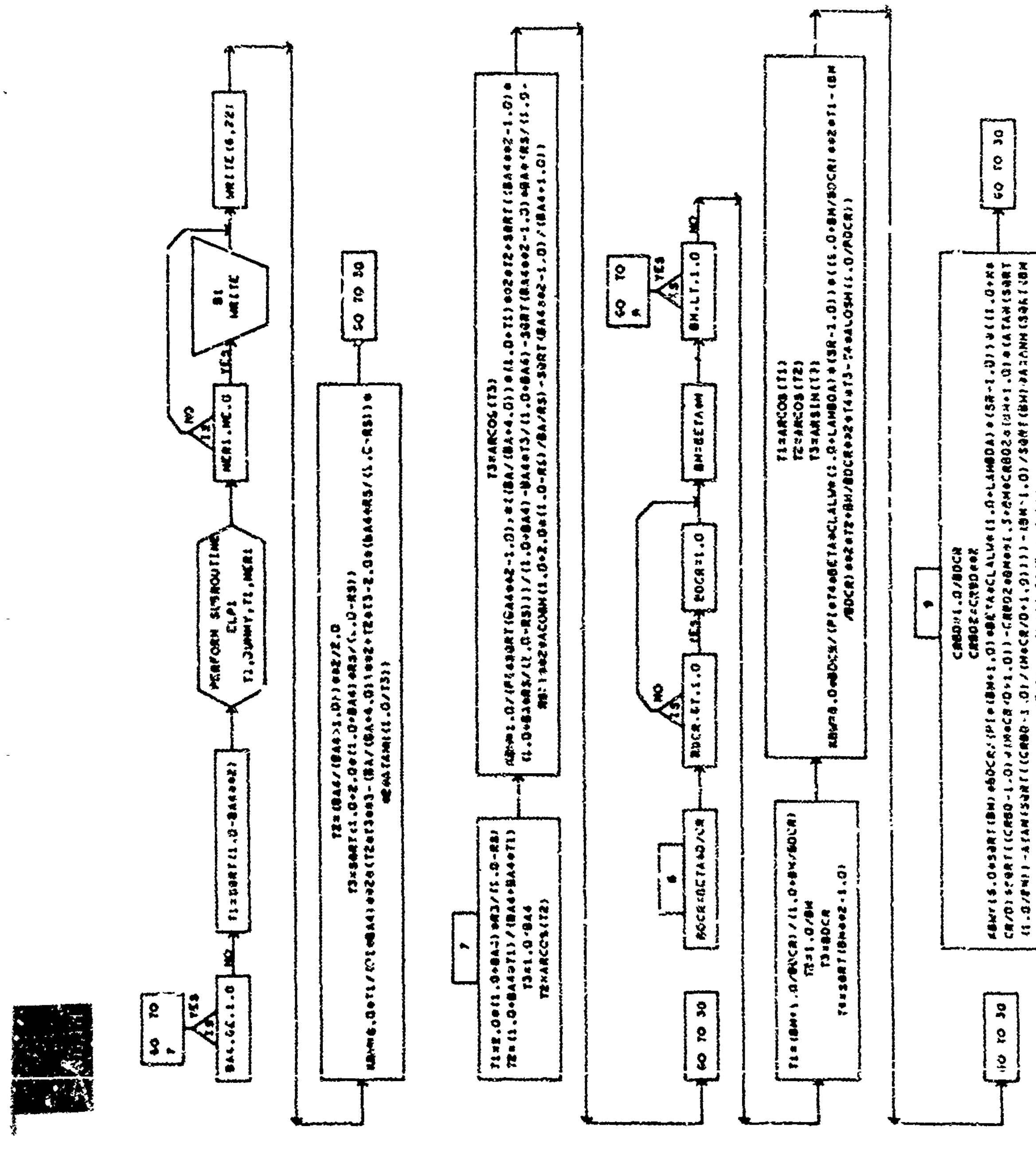
PLUNGE

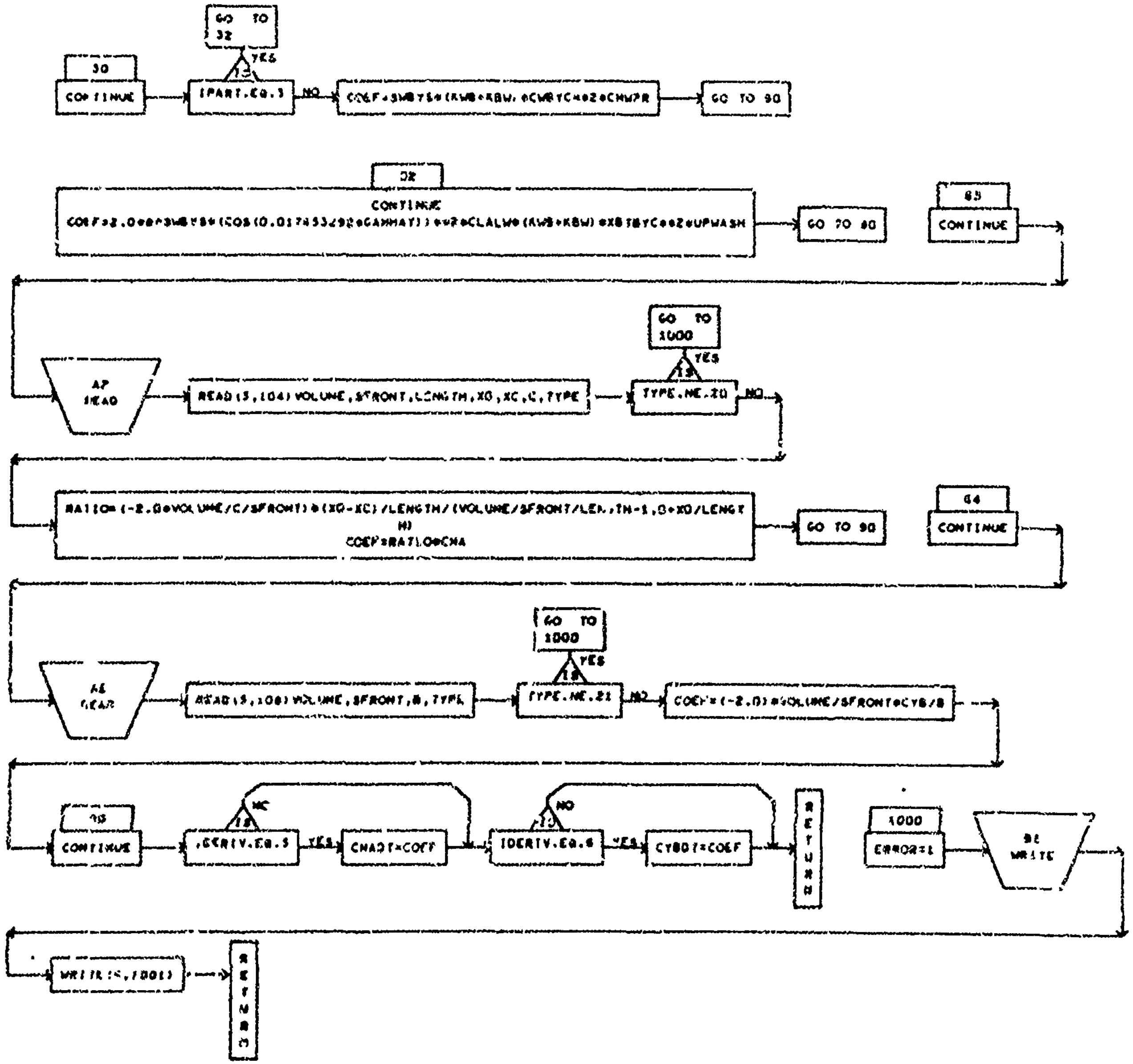
PLUNGE


TYPE.NE.15
IS
YES
GO TO 1000
NO
A2 READ
READ(5,104)CR,R,BETA,CLALW,CMWPR,KBW,TYPE
TYPE.NE.16
IS
YES
GO TO 1000
NO
GO TO 107
105 CONTINUE
A3 READ
READ(5,203)AR,LAMBDA,M,SIBYS,GAMMAT,S,K,TYPE
TYPE.NE.17
IS
YES
GO TO 1000
NO
A2 READ
READ(5,104)CR,R,BETA,CLALW,UPHASN,NB,BYC,TYPE
TYPE.NE.18
IS
YES
GO TO 1000
NO
A2 READ
READ(5,106)KBW,Q,TYPE
TYPE.NE.19
IS
YES
GO TO 1000
NO
107 CONTINUE
D=2.0*K
RS=R/S
SR=S/R
KWB=2.0/PI*((1.0+RS**4)*(0.5*ATAN(0.5*(SR-RS))+PI/4.0)-RS**2*(SR-RS+2.0*ATAN(RS)))/(1.0-RS)**2
ITYPE=K
ITYPE.GE.0.AND.ITYPE.LE.5
IS
YES
GO TO 72
NO
B1 WRITE
WRITE(6,71)ITYPE
GO TO 1000
72 CONTINUE
ITYPE.GT.1.AND.(BETA*AR*(1.0+LAMBDA)*(1.0/(BETA*M)+1.0)).LT.4.0
IS
YES
NO
ITYPE=1
I=ITYPE+1


COMPUTED GO TO
IF THE VALUE OF I IS — TRANSFER TO STATEMENT
1 — 30
2 — 2
3 — 5
4 — 5
5 — 6
6 — 3
2
KBW=((1.0-RS**2)**2-2.0/PI*((1.0+RS**4)*(0.5*ATAN(0.5*(SR-RS))+PI/4.0)-RS**2*(SR-RS+2.0*ATAN(RS))))/(1.0-RS)**2
GO TO 30
1
BH1=BETA*H
IS BH1.LT.1.0
YES
GO TO 4
NO
BH2=BH1**2
BHR2=SQRT(BH2-1.0)
T1=(1.0+(1.0+BH1)*BETA*O/CR)/(BH1+(BH1+1.0)*BETA*O/CR)
T2=1.0/BH1
T1=ARCOS(T1)
T2=ARCOS(T2)
KBW=8.0*BH1/(PI*BHR2*(1.0+LAMBDA)*BETA*O/CR*(SR-1.0)*BETA*CLALW)*((BH1/(1.0+BH1))**2*(((BH1+1.0)*BETA*O/CR+BH1)/BH1)**2*T1+BHR2/(BH1+1.0)*(SQRT(1.0+2.0*BETA*O/CR)-1.0)-BHR2/BH1*(BETA*O/CR)**2*ACOSH(1.0+CR/BETA/O)-BH1/(1.0+BH1)*T2)
GO TO 30
4
T1=SQRT((BH+(1.0+BH)*BETA*O/CR)/BH)
KBW=16.0*(BH/(1.0+BH))**2/(PI*(1.0+LAMBDA)*BETA*O/CR*(SR-1.0)*BETA*CLALW)*(T1**3+T1-2.0-(((1.0+BH)*BETA*O/CR)/BH)**2*ATANH(1.0/T1))
GO TO 30
5
BA=BETA*AR
BARS=BA*RS
T1=BA/(BA+SR-1.0)
T1=ARCOS(T1)
KBW=2.0/(PI*(BA-0.5))*(0.5*(1.0+BARS/(1.0-RS))**2*T1-0.5*(BARS/(1.0-RS))**2*ACOSH(1.0+(1.0-RS)/BARS)-0.5-PI/4.0+0.5*SQRT(1.0+2.0*BARS/(1.0-RS)))
GO TO 30
6
BA=BETA*AR
BA4=BA/4.0

PLUNGE





30
CONTINUE
IPART.EQ.3
YES
GO TO 32
NO
COEF=SWBYS*(KWB+KBW)*CWBYC*2*CHWPR
GO TO 90
32
CONTINUE
COEF=2.0*P*SWBYS*(COS(0.01745329*GAMMAY))**2*CLALW*(KWB+KBW)*XBTBYC*2*UPWASH
GO TO 80
83
CONTINUE
AP
READ
READ(5,104)VOLUME,SFRONT,LENGTH,XO,XC,C,TYPE
TYPE.NE.20
YES
GO TO 1000
NO
RATIO=(-2.0*VOLUME/C/SFRONT)*(XO-XC)/LENGTH/(VOLUME/SFRONT/LENGTH-1.0+XO/LENGTH)
COEF=RATIO*CNA
GO TO 90
84
CONTINUE
AE
READ
READ(5,108)VOLUME,SFRONT,B,TYPE
TYPE.NE.21
YES
GO TO 1000
NO
COEF=(-2.0)*VOLUME/SFRONT*CYB/B
90
CONTINUE
IDERIV.EQ.5
YES
NO
CHADT=COEF
IDERIV.EQ.6
YES
NO
CYBDT=COEF
RETURN
1000
ERROR=1
BE
WRITE
WRITE(6,1001)
RETURN

SYMBOLS USED IN SUBROUTINE PLUNGE

AR	R	U	ASPECT RATIO OF WING/TAIL	PLUNGE
B	R	U	WING/TAIL SPAN	PLUNGE
BA	R	U	PRODUCT OF BETA AND ASPECT RATIO	PLUNGE
BARS	R	U	PRODUCT OF BA AND RS	PLUNGE
BA4	R	U	BA DIVIDED BY 4	PLUNGE
BDCR	R	U	PRODUCT OF BETA AND D DIVIDED BY CR	PLUNGE
BETA	R	U	PRANDTL-GLAUERT FACTOR	PLUNGE
BM	R	U	PRODUCT OF BETA AND M	PLUNGE
BMR2	R	U	SQUARE ROOT OF DIFFERENCE OF BM2 AND 1.0	PLUNGE
BM1	R	U	PRODUCT OF BETA AND M	PLUNGE
BM2	R	U	SQUARE OF PRODUCT OF BETA AND M	PLUNGE
C	R	U	REFERENCE CHORD FOR BODY	PLUNGE
CASE	I	C	CASE NUMBER	PLUNGE
CLALW	R	U	LIFT-CURVE SLOPE FOR WING/TAIL (PER. RADIAN)	PLUNGE
CMA	R	A	DERIVATIVE OF PITCHING MOMENT WITH ANGLE OF ATTACK	PLUNGE
CMADT	R	A	PITCHING MOMENT-ALPHA DOT DERIVATIVE	PLUNGE
CMWPR	R	U	WING-ALONE/TAIL-ALONE PITCHING MOMENT DERIVATIVE	PLUNGE
COEF	R	U	COEFFICIENT DERIVATIVE	PLUNGE
CR	R	U	WING/TAIL CHORD AT WING/TAIL-BODY JUNCTURE	PLUNGE
CRBD	R	U	RICIPROCAL OF BDCR	PLUNGE
CRBD2	R	U	SQUARE OF CRBD	PLUNGE
CWBYC	R	U	MEAN AERODYNAMIC CHORD OF EXPOSED WING/TAIL	PLUNGE
CYB	R	A	DERIVATIVE OF SIDE FORCE WITH YAW ANGLE	PLUNGE
CYBDT	R	A	DERIVATIVE OF SIDE FORCE WITH BETA DOT	PLUNGE
D	R	U	BODY DIAMETER AT WING OR TAIL	PLUNGE
DUMMY	R	U	DUMMY VARIABLE	PLUNGE
ERROR	I	C	ERROR FLAG	PLUNGE
GAMMAT	R	U	TAIL DIHEDRAL ANGLE (DEGREES)	PLUNGE
I	I	U	INDEX	PLUNGE
ICALC	I	U	CALCULATION OPTION FLAG	PLUNGE
IDERIV	I	A	DERIVATIVE OPTION FLAG	PLUNGE
IPART	I	U	CONTROL FLAG FOR COMPONENT TYPE (BODY,WING,TAIL)	PLUNGE
ITYPE	I	U	FLAG TO CONTROL EQUATION TO BE USED IN CALCULATING KBW	PLUNGE
K	I	U	FLAG TO CONTROL SELECTION OF KBW EQUATION	PLUNGE
KBW	R	U	INTERFERENCE ON BODY IN PRESENCE OF WING/TAIL	PLUNGE
KWB	R	U	INTERFERENCE ON WING/TAIL IN PRESENCE OF BODY	PLUNGE

SYMBOLS USED IN SUBROUTINE PLUNGE

LAMBDA	R	U	WING/TAIL TAPER RATIO (TIP CHORD/ROOT CHORD)	PLUNGE
LENGTH	R	U	BODY LENGTH	PLUNGE
M	R	U	COTANGENT OF WING/TAIL LEADING EDGE SWEEP ANGLE	PLUNGE
NER1	I	U	ERROR FLAG	PLUNGE
PAGE	I	C	PAGE NUMBER	PLUNGE
PI	R	U	RATIO OF CIRCUMFERENCE OF A CIRCLE TO ITS DIAMETER	PLUNGE
Q	R	U	TAIL EFFECTIVENESS RATIO	PLUNGE
R	R	U	BODY RADIUS AT WING OR TAIL	PLUNGE
RATIO	R	U	DUMMY VARIABLE	PLUNGE
RS	R	U	R DIVIDED BY S	PLUNGE
S	R	U	WING/TAIL SEMI-SPAN	PLUNGE
SFRONT	R	U	BODY FRONTAL AREA	PLUNGE
SR	R	U	S DIVIDED BY R	PLUNGE
SWBYS	R	U	WING/TAIL AREA DIVIDED BY REFERENCE AREA	PLUNGE
TITLE	R	C	TITLE	PLUNGE
TYPE	I	U	CARD TYPE	PLUNGE
T1	R	U	DUMMY VARIABLE	PLUNGE
T2	R	U	DUMMY VARIABLE	PLUNGE
T3	R	U	DUMMY VARIABLE	PLUNGE
T4	R	U	DUMMY VARIABLE	PLUNGE
UPWASH	R	U	TAIL UPWASH DERIVATIVE CAUSED BY WING	PLUNGE
VOLUME	R	U	BODY VOLUME	PLUNGE
XBTBYC	R	U	TAIL LENGTH DIVIDED BY REFERENCE CHORD	PLUNGE
XC	R	U	AREA CENTROID LOCATION OF BODY	PLUNGE
XO	R	U	CENTER OF GRAVITY LOCATION	PLUNGE

26. SUBROUTINE ELP1 (DECK AROY)

This routine approximates the values of the elliptical integrals of the first and second kinds.

a. Algorithm

The approximation to a value of the elliptical integral of the first kind is given by

$$K(k) = (a_o + a_1\eta + \ldots + a_4\eta^4) + (b_o + b_1\eta + \ldots + b_4\eta^4) \ln \frac{1}{\eta}$$

where $\eta = 1 - k^2$

$a_o = 1.386294361$	$b_o = 0.5$
$a_1 = 0.0966634426$	$b_1 = 0.124985936$
$a_2 = 0.0359009238$	$b_2 = 0.0688024857$
$a_3 = 0.0374256371$	$b_3 = 0.0332835534$
$a_4 = 0.0145119621$	$b_4 = 0.0044178701$

The approximation to the value of the elliptical integral of the second kind is given by

$$E(k) = (a_o + a_1\eta + \ldots + a_4\eta^4) + (b_o + b_1\eta + \ldots + b_4\eta^4) \ln \frac{1}{\eta}$$

where $\eta = 1 - k^2$

$a_o = 1.0$	$b_o = 0.0$
$a_1 = 0.4432514146$	$b_1 = 0.2499836831$
$a_2 = 0.0626060122$	$b_2 = 0.0920018004$
$a_3 = 0.0475738355$	$b_3 = 0.0406969753$
$a_4 = 0.0173650645$	$b_4 = 0.0052644964$

b. Input/Output

None

c. Error

None

d. Subroutines Required

None

e. Argument List

(AK, AKK, E, NERROR)

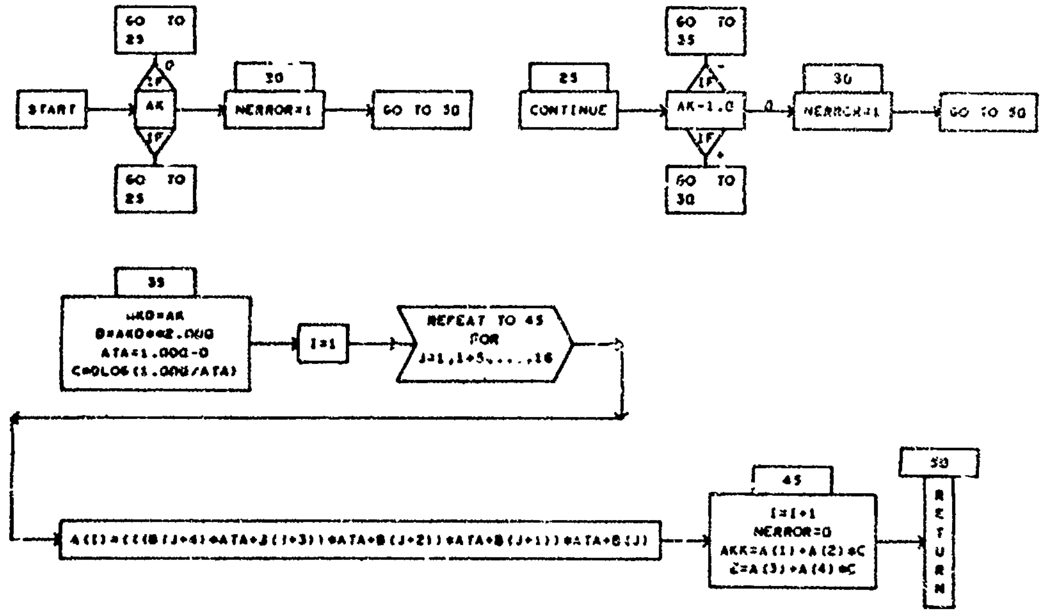
f. Length

828 bytes

```
DECK AROY

      SUBROUTINE ELP1(AK, AKK, E, NERROR)
CELP1  ELP1 IS A ROUTINE TO APPROXIMATE THE VALUE OF THE ELLIPTICAL
C     INTEGRALS E(K) AND K(K) BY A METHOD OF NUMERICAL ANALYSIS.
C     ARGUMENTS OF THE ROUTINE ARE AS FOLLOWS
C     AK    = INPUT VALUE OF K
C     AKK   = OUTPUT VALUE OF K(K) ELLIPTICAL INTEGRAL
C     F     = OUTPUT VALUE OF E(K) ELLIPTICAL INTEGRAL
C     NERROR = ERROR CODE
C              0 = K IS IN ALLOWABLE RANGE
C              1 = K OUT OF ALLOWABLE RANGE, I.E. EQUAL TO OR GREATER
C                  THAN ONE OR LESS THAN ZERO.
C
      DIMENSION B(20), A(4)
      DOUBLE PRECISION A, B, ATA, C, D, AKD
      DATA                  B/1.38629436112, 0.09666344259, 0.03590092383,
     1
     2 .03742563713, .01451196212, .5D0, .12498593597, .06880248576,
     3
     4 .03328355346, .00417870120D0, 1.0D0, .44325141463, .06260601220,
     5
     6 .04757383546, .01736506451, .0D0, .29499836831, .09200180037,
     7
     8 .04069697526, .00526449639D0/
C
      IF(AK) 20,25,25
   20 NERROR = 1
      GO TO 50
   25 IF(AK-1.0) 35,30,30
   30 NERROR = 1
      GO TO 50
   35 AKD = AK
      D = AKD ** 2.0D0
      ATA = 1.0D0 - D
      C = DLOG(1.0D0 / ATA)
      I = 1
```

```
DECK ARDY

      DO 45 J = 1,16,5                                                  ARDY  0360
      A(I) = (((B(J+4)*ATA + B(J+3))*ATA + B(J+2))*ATA + B(J+1))*ATA +  ARDY  0370
     1      B(J)                                                        ARDY  0380
   45 I = I + 1                                                         ARDY  0390
      NERROR = 0                                                        ARDY  0400
      AKK = A(1) + A(2)*C                                               ARDY  0410
      F   = A(3) + A(4)*C                                               ARDY  0420
   50 RETURN                                                            ARDY  0430
      END                                                               ARDY  0440
```

SUBROUTINE ELP1



SYMBOLS USED IN SUBROUTINE ELP1

A	D	D	ELLIPTICAL INTEGRAL PARAMETER	ELP1
AK	R	A	THE MODULUS	ELP1
AKO	D	U	MODULUS	ELP1
AKK	R	A	ELLIPTICAL INTEGRAL OF THE FIRST KIND	ELP1
ATA	D	U	ELLIPTICAL INTEGRAL PARAMETER	ELP1
B	D	D	ELLIPTICAL INTEGRAL CONSTANT ARRAY	ELP1
C	D	U	ELLIPTICAL INTEGRAL PARAMETER	ELP1
D	D	U	ELLIPTICAL INTEGRAL PARAMETER	ELP1
E	R	A	ELLIPTICAL INTEGRAL OF THE SECOND KIND	ELP1
I	I	U	INDEX	ELP1
J	I	U	DO-LOOP INDEX	ELP1
NERROR	I	A	ERROR FLAG	ELP1

27. SUBROUTINE VECTOR (DECK AROZ)

This routine converts input thrust vector data to coefficients and adds the results to the vehicle aerodynamic coefficients.

a. Algorithm

Set up initial conditions and constants. Read in a force vector and convert to force coefficients. Print the results if required. Continue to read in force vector data until LAST equals one.

b. Input/Output

Thrust Vector Data Cards (Type 22)
Thrust Vector coefficient contributions are printed if IPRINT = 1.

c. Error

An error condition occurs if the card type number is wrong.

d. Subroutines Required

HEADER

e. Argument List

(MACH, PFS, SREF, XCG, YCG, ZCG, SPAN, MAC, ALPHA, CD, CL, CA, CY, CN, BETA, LOD, CLM, CLL, CLN)

f. Length

2304 bytes

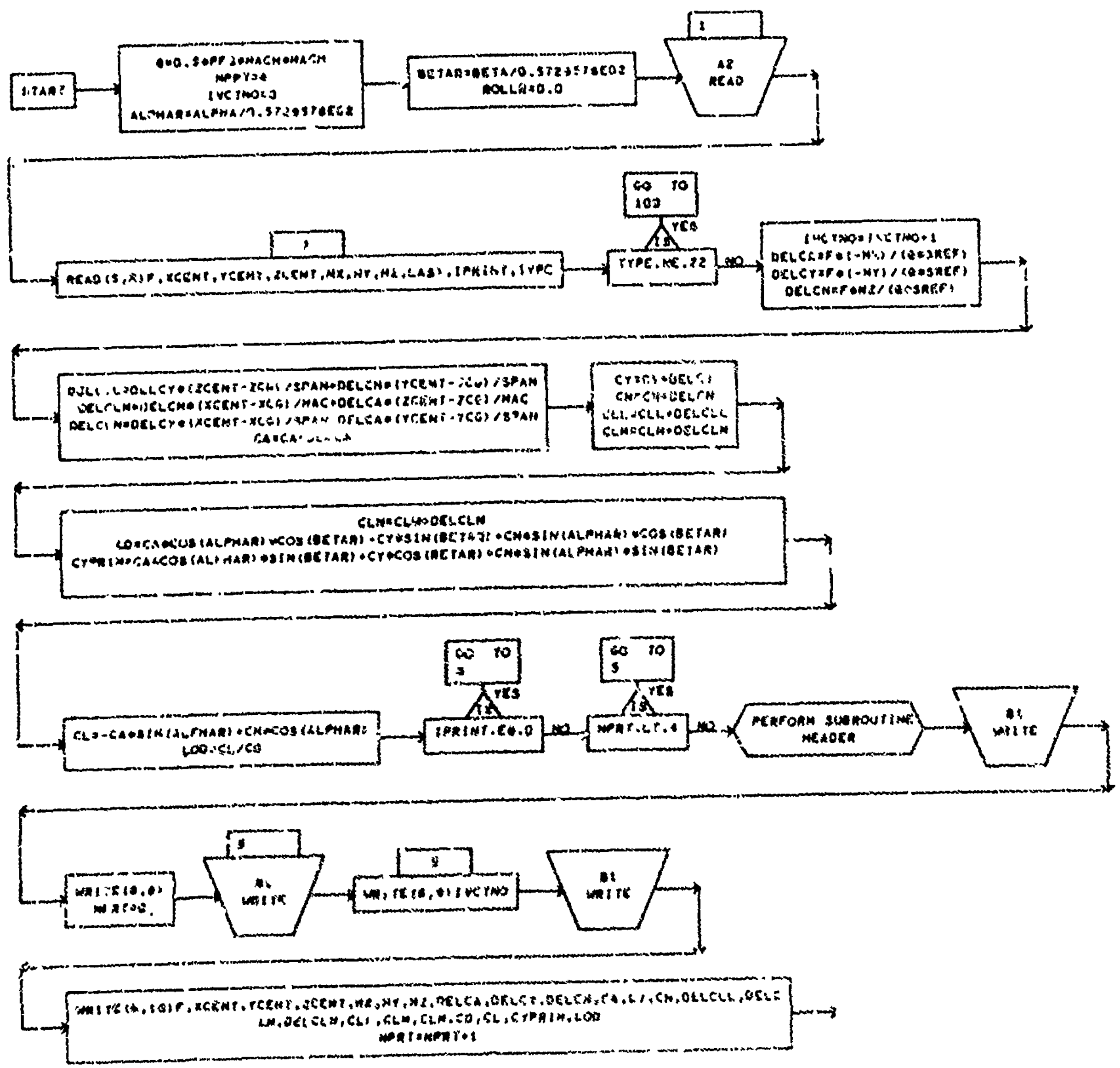
DECK AROZ

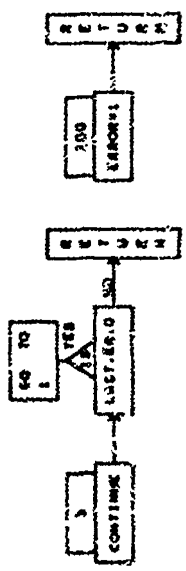
```
      SUBROUTINE VECTOR (MACH,PFS,SREF,XCG,YCG,ZCG,SPAN,MAC,
     1 ALPHA,CD,CL,CA,CY,CN,BETA,LOD,CLM,CLL,CLN)
C
C***********************************************************************
C**** THIS SUBROUTINE CONVERTS THRUST VECTORS INTO AERODYNAMIC
C**** COEFFICIENTS.  ANY NUMBER OF VECTORS MAY BE INPUT.
C***********************************************************************
C
      DIMENSION  TITLE(15)
      COMMON  CASE,TITLE,PAGE,ERROR
      REAL  MACH,MAC,LOD,NX,NY,NZ
      INTEGER  ERROR,CASE,PAGE,TYPE
C
C  SET UP INITIAL CONDITIONS
      Q = 0.5 * PFS * MACH*MACH
      NPRT = 4
      IVCTNO = 0
      ALPHAR = ALPHA / 0.5729578E02
      BETAR  = BETA  / 0.5729578E02
      ROLLR = 0.0
C
C  READ IN FORCE VECTOR DATA AND CONVERT TO FORCE COEFFICIENTS
    1 READ (5,2) F,XCENT,YCENT,ZCENT,NX,NY,NZ,LAST,IPRINT,TYPE
    2 FORMAT (F10.0,6F6.0,13X,I1,1XI1,8X,I2)
      IF (TYPE .NE. 22) GO TO 100
      IVCTNO = IVCTNO + 1
      DELCA = F *(-NX)/ (Q * SREF)
      DELCY = F *(-NY)/ (Q * SREF)
      DELCN = F *  NZ / (Q * SREF)
      DELCLL= DELCY * (ZCENT - ZCG)/SPAN
     1       +DELCN * (YCENT - YCG)/SPAN
      DELCLM= DELCN * (XCENT - XCG)/MAC
     1       +DELCA * (ZCENT - ZCG)/MAC
      DELCLN= DELCY * (XCENT - XCG)/SPAN
     1       -DELCA * (YCENT - YCG)/SPAN
```

```
DECK AROZ

      CA   = CA   + DELCA
      CY   = CY   + DELCY
      CN   = CN   + DELCN
      CLL  = CLL  + DELCLL
      CLM  = CLM  + DELCLM
      CLN  = CLN  + DELCLN
      CD = CA*COS(ALPHAR)*COS(BETAR) - CY*SIN(BETAR)
     1    +CN*SIN(ALPHAR)*COS(BETAR)
      CYPRIM = CA*COS(ALPHAR)*SIN(BETAR) + CY*COS(BETAR)
     1         +CN*SIN(ALPHAR)*SIN(BETAR)
C
C
      CL = -CA*SIN(ALPHAR) + CN*COS(ALPHAR)
C
C
C
      LOD = CL / CD
C
C    PRINT RESULTS IF REQUIRED
      IF (IPRINT .EQ. 0) GO TO 3
      IF (NPRT .LT. 4) GO TO 5
      CALL HEADER
      WRITE (6,8)
    8 FORMAT (1H0,50HRESULTS OF VECTOR CONVERSION TO FORCE COEFFICIENTS)
      NPRT = 0
    5 WRITE (6,9) IVCTNO
    9 FORMAT (1H0,13HVECTOR NUMBER,I3)
      WRITE (6,10) F,XCENT,YCENT,ZCENT,NX,NY,NZ,DELCA,DELCY,DELCN,
     1  CA,CY,CN,DELCLL,DELCLM,DELCLN,CLL,CLM,CLN,CD,CL,CYPRIM,LOD
   10 FORMAT (1H ,5X,3HF =F12.1,2X6HXCENT=F7.1,2X6HYCENT=F7.1,
     1  2X6HZCENT=F7.1,/1H ,23X3HNX=F7.4,5X3HNX=F7.4,5X3HNX=F7.4,/
     2  1H ,3X8HDEL CA =E12.5,2X8HDEL CY =E12.5,2X8HDEL CN =E12.5,/
     3  1H ,3X8HTOT CA =E12.5,2X8HTOT CY =E12.5,2X8HTOT CN =E12.5,/
     4  1H0,3X8HDEL CLL=E12.5,2X8HDEL CLM=E12.5,2X8HDEL CLN=E12.5,/
     5  1H ,3X8HTOT CLL=E12.5,2X8HTOT CLM=E12.5,2X8HTOT CLN=E12.5,/
     6  1H0,6X5HC D =F9.5,8X5HC L =F9.5,7X,5HC Y =F9.5,/
```

```
DECK AROZ
                                                         AROZ  0720
     7  1H ,6X,5HL/D =F9.3)                              AROZ  0730
        NPRT = NPRT + 1                                  AROZ  0740
   3    IF (LAST .EQ. 0) GO TO 1                         AROZ  0750
        RETURN                                           AROZ  0760
 100    ERROR = 1                                        AROZ  0770
        RETURN                                           AROZ  0780
        END
```

SUBROUTINE VECTOR




RETURN
200
RETURN
GO TO
YES
NO
CONTINUE

SYMBOLS USED IN SUBROUTINE VECTOR

ALPHA	R	A	ANGLE OF ATTACK, DEGREES	VECTOR
ALPHAR	R	U	ANGLE OF ATTACK, RADIANS	VECTOR
BETA	R	A	YAW ANGLE, DEGREES	VECTOR
BETAR	R	U	YAW ANGLE, RADIANS	VECTOR
CA	R	A	AXIAL FORCE COEFFICIENT	VECTOR
CASE	I	C	CASE NUMBER	VECTOR
CD	R	A	DRAG COEFFICIENT	VECTOR
CL	R	A	LIFT COEFFICIENT	VECTOR
CLL	R	A	ROLLING MOMENT COEFFICIENT	VECTOR
CLM	R	A	PITCHING MOMENT COEFFICIENT	VECTOR
CLN	R	A	YAWING MOMENT COEFFICIENT	VECTOR
CN	R	A	NORMAL FORCE COEFFICIENT	VECTOR
CY	R	A	SIDE FORCE COEFFICIENT	VECTOR
CYPRIM	R	U	SIDE FORCE COEFFICIENT (WIND AXIS)	VECTOR
DELCA	R	U	DRAG COEFFICIENT INCREMENT	VECTOR
DELCLL	R	U	ROLLING MOMENT COEFFICIENT INCREMENT	VECTOR
DELCLM	R	U	PITCHING MOMENT COEFFICIENT INCREMENT	VECTOR
DELCLN	R	U	YAWING MOMENT COEFFICIENT INCREMENT	VECTOR
DELCN	R	U	NORMAL FORCE COEFFICIENT INCREMENT	VECTOR
DELCY	R	U	SIDE FORCE COEFFICIENT INCREMENT	VECTOR
ERROR	I	C	ERROR FLAG	VECTOR
F	R	U	FORCE MAGNITUDE, POUNDS	VECTOR
IPRINT	I	U	PRINT FLAG	VECTOR
IVCTNU	I	U	VECTOR NUMBER	VECTOR
LAST	I	U	LAST VECTOR FLAG	VECTOR
LOD	R	A	LIFT-TO-DRAG RATIO	VECTOR
MAC	R	A	REFERENCE LENGTH FOR MOMENT COEFFICIENTS	VECTOR
MACH	R	A	MACH NUMBER	VECTOR
NPRT	I	U	PRINT LINE COUNTER	VECTOR
NX	R	U	FORCE VECTOR DIRECTION COSINE IN X-DIRECTION	VECTOR
NY	R	U	FORCE VECTOR DIRECTION COSINE IN Y-DIRECTION	VECTOR
NZ	R	U	FORCE VECTOR DIRECTION COSINE IN Z-DIRECTION	VECTOR
PAGE	I	C	PAGE NUMBER	VECTOR
PFS	R	A	FREE-STREAM PRESSURE, LBS/SQUARE FOOT	VECTOR
Q	R	U	DYNAMIC PRESSURE	VECTOR
ROLLR	R	U	ROLL ANGLE IN RADIANS	VECTOR

SYMBOLS USED IN SUBROUTINE VECTOR

```
SPAN     R  A  REFERENCE LENGTH FOR ROLLING , YAWING COEFFICIENTS      VECTOR
SREF     R  A  VEHICLE REFERENCE AREA (WING AREA)                      VECTOR
TITLE    R  C  TITLE                                                   VECTOR
TYPE     I  U  CARD TYPE NUMBER                                        VECTOR
XCENT    R  U  ACTION POINT FOR FORCE VECTOR-X                         VECTOR
XCG      R  A  X-CENTER FOR MOMENT CALCULATIONS                        VECTOR
YCENT    R  U  ACTION POINT FOR FORCE VECTOR-Y                         VECTOR
YCG      R  A  Y-CENTER FOR MOMENT CALCULATIONS                        VECTOR
ZCENT    R  U  ACTION POINT FOR FORCE VECTOR-Z                         VECTOR
ZCG      R  A  Z-CENTER FOR MOMENT CALCULATIONS                        VECTOR
```

28. SUBROUTINE GRAPIC (DECK GRPA)

This is the Executive routine for the graphics part of the program.

a. Algorithm

Print that GRAPHIC OPTION HAS CONTROL and select the proper graphic routine depending upon the value of IPROG.

b. Input/Output

None

c. Error

None

d. Subroutines Required

PICTUR, PLOT

e. Argument List

(IPROG)

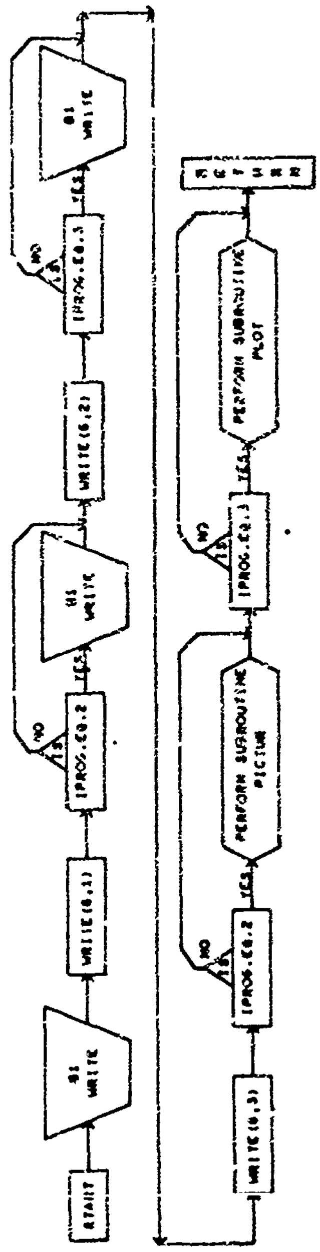
f. Length

536 bytes

DECK GRPA

```
      SUBROUTINE GRAPIC (IPROG)                                         GRPA 0010
C                                                                       GRPA 0020
      COMMON CASE,TITLE,PAGE,ERROR                                      GRPA 0030
      DIMENSION  TITLE(15)                                              GRPA 0040
      INTEGER  ERROR,CASE,PAGE                                          GRPA 0050
C                                                                       GRPA 0060
C***** GRAPHIC OUTPUT DATA CONTROL PROGRAM *******************           GRPA 0070
C                                                                       GRPA 0080
C                                                                       GRPA 0090
      WRITE (6,1)                                                       GRPA 0100
    1 FORMAT (1H ,///,1H ,38H***** GRAPHIC OPTION HAS CONTROL ***** )     GRPA 0110
      IF (IPROG .EQ. 2) WRITE (6,2)                                     GRPA 0120
    2 FORMAT (1HO,10X,40HPICTURE DRAWING PROGRAM WILL BE EXECUTED  )     GRPA 0130
      IF (IPROG .EQ. 3) WRITE (6,3)                                     GRPA 0140
    3 FORMAT (1HO,10X,44HOUTPUT DATA PLOTTER PROGRAM WILL BE EXECUTED )  GRPA 0150
C                                                                       GRPA 0160
      IF (IPROG.EQ. 2) CALL PICTUR                                      GRPA 0170
C                                                                       GRPA 0180
      IF (IPROG .EQ. 3) CALL PLOT                                       GRPA 0190
C                                                                       GRPA 0200
      RETURN                                                            GRPA 0210
      END                                                               GRPA 0220
```

SUBROUTINE GRAPIC



SYMBOLS USED IN SUBROUTINE GRAPIC

CASE	I	C	CASE NUMBER	GRAPIC
ERROR	I	C	ERROR FLAG	GRAPIC
IPROG	I	A	PROGRAM OPTION NUMBER	GRAPIC
PAGE	I	C	PAGE NUMBER	GRAPIC
TITLE	R	C	TITLE	GRAPIC

29. SUBROUTINE PICTUR (DECK GRPB)

This routine prepares an output tape for procession on the SC-4020. The result will be pictures of the vehicle with the selected viewing angles.

a. Algorithm

Read the Picture Drawing Program Element Data Title Card (Type 31) and the Element Data Control Card (Type 32). Read the surface element data either from Tape 5 or from Tape 8 as directed. Read plotting instruction data (Card Types 34, 35, 36, and 37). Set up starting constants for pictures. Read element data from Tape 3 using the same techniques as for SDATA and convert to quadrilaterals. Generate scale grids if required. Plot points and draw lines between the points as directed by the input data. Print the detailed element characteristics if PRINTS is equal to 1.

b. Input/Output

Element Data Title Card (Type 31), Element Data Control Card (Type 32), Element Data Cards from Tape 5 or Tape 8 (Type 3), Picture Control Data Card (Type 34), Grid Data Card (Type 35), Scale Label Card (Type 36), and Plot Title Card(s) (Type 37). If PRINTS is equal to 1 the detailed element characteristics will be printed on Tape 6 (just as is the case for SDATA).

c. Error

An error condition occurs if a card type number is wrong.

d. Subroutines Required

HEADR2, SC-4020 routines

e. Argument List

None

f. Length

16180 bytes

DECK GRPB

```
      SUBROUTINE PICTUR
C      SC-4020 PLOTTER PROGRAM FOR PLOTTING SURFACE DATA
C
C
      DIMENSION   XA(250),XB(250),YA(250),YB(250),ZA(250),ZB(250),
     1 XI(4),ETA(4),XIN(4),YIN(4),ZIN(4),VTITLE(8),HTITLE(8),HLABEL(15),
     2 YIN2(4),ZIN2(4),CARD(20),TITLE(15)
C
      COMMON  CASE,TITLE,PAGE,ERROR
C
      REAL   NX,NY,NZ,NXO
      LOGICAL   RFLAG, AFLAG, BFLAG
      INTEGER   STAT,STATT,TYPE,PRINTS,SYMFCT,CASE,PAGE,ERROR
C
C
      FIRST(QX,QY,QZ,Q1,Q2,Q3) = QX*Q1 + QY*Q2 + QZ*Q3
C
      THIRD(QX,QY,QZ,QPSI,QTHETA,QPHI) = QX*(COS(QTHETA)*COS(QPSI)) +
     1  QY*(-SIN(QPSI)*COS(QPHI)+SIN(QTHETA)*COS(QPSI)*SIN(QPHI)) +
     2  QZ*(SIN(QPSI)*SIN(QPHI)+SIN(QTHETA)*COS(QPSI)*COS(QPHI))
C
      CALL CAMRAV (9)
      IDUM = -1
      REWIND 2
      CALL INCRV (6,3)
      IPIC = 1
    2 REWIND 3
C  READ ALL INPUT DATA AND STORE ON TAPE 3 FOR FUTURE USE
      READ(5,100) (TITLE(I),I=1,15),CASE,TYPE
  100 FORMAT(14A4,A3,6X,I3,2XI2)
      IF (TYPE .NE. 99) GO TO 301
      ERROR = 0
      RETURN
  301 IF (TYPE .NE. 51) GO TO 300
      READ(5,101) PRINTS,SYMFCT,IORIEN,IFACT,XSC,YSC,ZSC,DELX,DELY,DELZ,
```

```
DECK GRPB

    1      ISTAT3,ITAPE,IREW8,TYPE                                        GRPB  0360
 101  FORMAT (I1,1XI1,2I1,1X3F6.0,1X3F6.0,16XI2,2I1,7XI2)                 GRPB  0370
      IF (TYPE .NE. 32) GO TO 300                                          GRPB  0380
      IF (IREW8 .EQ. 0) REWIND 8                                           GRPB  0390
      IF (SYMFCT .EQ. 0) SYMFCT = 2                                        GRPB  0400
      IA = 0                                                               GRPB  0410
      IB = 0                                                               GRPB  0420
      IF (IORIEN .EQ. 2) IB = 1                                            GRPB  0430
      IF (IORIEN .EQ. 3) IA = 1                                            GRPB  0440
    3 IF (ITAPE.EQ.0) READ (5,1) X,Y,Z,STAT,XX,YY,ZZ,STATT,TYPE           GRPB  0450
      IF (ITAPE.NE.0) READ (8,1) X,Y,Z,STAT,XX,YY,ZZ,STATT,TYPE           GRPB  0460
    1 FORMAT (2(3F10.0,I1),8XI2 )                                          GRPB  0470
      IF (TYPE.NE.3 .AND. ITAPE.EQ.0) GO TO 300                            GRPB  0480
      IF (TYPE.NE.3 .AND. ITAPE.NE.0) GO TO 806                            GRPB  0490
C                                                                          GRPB  0500
      IF (IFACT.EQ.0) GO TO 10                                             GRPB  0510
        X = X * XSC + DELX                                                 GRPB  0520
        XX= XX* XSC + DELX                                                 GRPB  0530
        Y = Y * YSC + DELY                                                 GRPB  0540
        YY= YY* YSC + DELY                                                 GRPB  0550
        Z = Z * ZSC + DELZ                                                 GRPB  0560
        ZZ= ZZ* ZSC + DELZ                                                 GRPB  0570
  10  IF (STAT.EQ.3 .OR. STATT.EQ.3 .AND. ISTAT3.GT.0) ISTAT3 = ISTAT3-1  GRPB  0580
      IF (STAT.EQ.3 .AND. ISTAT3.GT.0) STAT = 0                            GRPB  0590
      IF (STATT.EQ.3 .AND. ISTAT3.GT.0) STATT = 0                          GRPB  0600
      WRITE (3) X,Y,Z,STAT,XX,YY,ZZ,STATT                                  GRPB  0610
      IF (STAT.EQ.3 .OR. STATT.EQ.3) GO TO 4                               GRPB  0620
      GO TO 3                                                              GRPB  0630
    4 WRITE (6,4016) IPIC                                                  GRPB  0640
 4016 FORMAT (1H1,////,1H0,34HPICTURE DRAWING PROGRAM     PICTURE          GRPB  0650
    1      7H NUMBER ,I4)                                                  GRPB  0660
      IPIC = IPIC + 1                                                      GRPB  0670
C                                                                          GRPB  0680
C  READ PLOTTING INSTRUCTIONS                                              GRPB  0690
      READ (5,5) PSI,THETA,PHI,ICS,IREFL,ISHAD,IAREA,IQUAD,IFRAME,NCAM,    GRPB  0700
    1   MARKPT,NG,MG,IG,JG,NXG,NYG,LAST,TYPE                               GRPB  0710
```

```
DECK GRPB
    5 FORMAT (F6.0,1XF6.0,1XF6.0,6(1XI1),2(1XI2),2X2I3,1X2I3,1X2I2,
   1  1XI1,10XI2 )
      IF (TYPE .NE. 34) GO TO 300
      READ (5,6) XLG,XRG,YBG,YTG,DXG,DYG,NOSCAL,TYPE
    6 FORMAT (5F10.0,F9.0,I1,10XI2)
      IF (TYPE .NE. 35) GO TO 300
      IF (NOSCAL .EQ. 1) GO TO 8
      READ (5,7) (VTITLE(I),I=1,8),(HTITLE(I),I=1,8),TYPE
    7 FORMAT (7A4,A2,7A4,A1,11XI2)
      IF (TYPE .NE. 36) GO TO 300
C
    8 CALL CAMRAV (NCAM)
      CALL FRAMEV (0)
C
C     SET UP STARTING CONSTANTS
      IFADV = 1
      ISTART = 0
      PSI   =  PSI   / 57.2957795
      THETA =  THETA / 57.2957795
      PHI   =  PHI   / 57.2957795
      SINTH = SIN(THETA)
      COSTH = COS(THETA)
      SINPSI= SIN(PSI)
      COSPSI= COS(PSI)
      SINPHI= SIN(PHI)
      COSPHI= COS(PHI)
      A1 = COSTH * SINPSI
      A2 = COSPSI*COSPHI + SINTH*SINPSI*SINPHI
      A3 =-COSPSI*SINPHI + SINTH*SINPSI*COSPHI
      A4 =-SINTH
      A5 = COSTH*SINPHI
      A6 = COSTH*COSPHI
      A7 = COSTH*COSPSI
      A8 =-SINPSI*COSPHI + SINTH*COSPSI*SINPHI
      A9 = SINPSI*SINPHI + SINTH*COSPSI*COSPHI
       N = -1
```

```
DECK GRPB

      NK = - 1
       KLCT = 0
       L = 0
       NPRT = 10
       AREAT = 0.0
       VOL = 0.0
      REWIND 3
C
C
C     READ IN ALL SURFACE DATA
   29 READ (3) X,Y,Z,STAT, XX,YY,ZZ,STATT
       RFLAG = .FALSE.
       GO TO 80
   30 IF (RFLAG) GO TO 50
       RFLAG = .TRUE.
       X = XX
       Y = YY
       Z = ZZ
       STAT = STATT
       GO TO 60
   50 RFLAG = .FALSE.
      READ (3) X,Y,Z,STAT, XX,YY,ZZ,STATT
   60 IF (STAT .EQ. 0 .OR. STAT .EQ. 3) GO TO 180
      IF (STAT .EQ. 2) GO TO 200
   70 IF (.NOT. AFLAG) GO TO 200
       MC = M
   80 K = 1
       IF (STAT .EQ. 2) GO TO 150
       IF (.NOT. BFLAG) GO TO 84
   75 DO 81 J =1,MC
         XA(J) = XB(J)
         YA(J) = YB(J)
   81    ZA(J) = ZB(J)
   83 XB(1) = X
      YB(1) = Y
      ZB(1) = Z
```

```
DECK GRPB

      GO TO 30
   34 IF (AFLAG) GO TO 85
       BFLAG = .TRUE.
       GO TO 75
   85 AFLAG = .FALSE.
       GO TO 83
  150 AFLAG = .TRUE.
       BFLAG = .FALSE.
       N = N+1
  160 XA(M) = X
      YA(M) = Y
      ZA(M) = Z
      GO TO 30
  180 M = M + 1
      IF (AFLAG) GO TO 160
       XB(N) = X
       YB(N) = Y
       ZB(N) = Z
      IF (STAT .NE. 3) GO TO 30
  200 MMIN = MINO (M,MC) - 1
       NN2 = 1
       MC = M
  250 N = N + 1
      NN = NN + 1
       KLCT = KLCT + 1
C
C
C  BEGIN COMPUTATION OF SURFACE ELEMENT CHARACTERISTICS
C
  450 DO 2000 I= 1,MMIN
      IIA = I + IA
      IIB = I + IB
        XIN(1) = XA(IIA )
        XIN(2) = XA(IIA +1)
        XIN(3) = XB(IIB +1)
        XIN(4) = XB(IIB )
```

```
DECK GRPB
      YIN(1) = YA(IIA )
      YIN(2) = YA(IIA +1)
      YIN(3) = YB(IIB +1)
      YIN(4) = YB(IIB )
      ZIN(1) = ZA(IIA )
      ZIN(2) = ZA(IIA +1)
      ZIN(3) = ZB(IIB +1)
      ZIN(4) = ZB(IIB )
      IRFLG = 0
C
C  FORM DIAGONAL VECTORS
      T1X = XIN(3) - XIN(1)
      T2X = XIN(4) - XIN(2)
      T1Y = YIN(3) - YIN(1)
      T2Y = YIN(4) - YIN(2)
      T1Z = ZIN(3) - ZIN(1)
      T2Z = ZIN(4) - ZIN(2)
C
C  FORM CROSS PRODUCT  N=T2 X T1
      NX = T2Y*T1Z - T1Y*T2Z
      NY = T1X*T2Z - T2X*T1Z
      NZ = T2X*T1Y - T1X*T2Y
      VN = SQRT ( NX*NX + NY*NY + NZ*NZ )
C
C  FORM UNIT NORMAL VECTOR
      IF (VN .EQ. 0.0) GO TO 421
      NX = NX / VN
      NY = NY / VN
      NZ = NZ / VN
C
C  COMPUTE AVERAGE POINT
 421  AVX = 0.25 * (XIN(1) + XIN(2) + XIN(3) + XIN(4) )
      AVY = 0.25 * (YIN(1) + YIN(2) + YIN(3) + YIN(4) )
      AVZ = 0.25 * (ZIN(1) + ZIN(2) + ZIN(3) + ZIN(4) )
C
C  COMPUTE PROJECTION DISTANCE
```

```
DECK GRPB

      D = NX*(AVX - XIN(1)) + NY*(AVY - YIN(1)) + NZ*(AVZ-ZIN(1))
      PD = ABS(D)
C
C
      T = SQRT (T1X*T1X + T1Y*T1Y + T1Z*T1Z)
     IF (T .EQ. 0.0) GO TO 431
     T1X = T1X / T
      T1Y = T1Y / T
      T1Z = T1Z / T
C
C
 431 T2X = NY*T1Z - NZ*T1Y
      T2Y = NZ*T1X - NX*T1Z
      T2Z = NX*T1Y - NY*T1X
C
C  COMPUTE COORDINATES OF CORNER POINTS IN REFERENCE COORD. SYSTEM
C
     DO 1000 J = 1,4
      XPA= XIN(J) + NX*D
      YPA= YIN(J) + NY*D
      ZPA= ZIN(J) + NZ*D
C
     IF (IQUAD .EQ. 0) GO TO 470
       XIN(J) = XPA
       YIN(J) = YPA
       ZIN(J) = ZPA
C
 470  D = - D
      XDIF = XPA- AVX
      YDIF = YPA- AVY
      ZDIF = ZPA- AVZ
C
C  TRANSFORM CORNER POINTS TO ELEMENT COORDINATE SYSTEM (XI,ETA) WITH
C   AVERAGE POINT AS ORIGIN
      XI(J) = T1X*XDIF + T1Y*YDIF + T1Z*ZDIF
 1000 ETA(J)= T2X*XDIF + T2Y*YDIF + T2Z*ZDIF
```

```
DECK GRPB
C
C  COMPUTE CENTROID
      ETACK = ETA(2) - ETA(4)
      IF (ETACK .NE. 0.0) GO TO 432
       XIO = 0.0
       GO TO 433
 432  XIO = .333333333 * (XI(4) * (ETA(1)-ETA(2)) + XI(2)
     1       * (ETA(4)-ETA(1))) / (ETA(2)-ETA(4))
 433  ETAO = -.333333333 * ETA(1)
C
C  OBTAIN CORNER POINTS IN SYSTEM WITH CENTROID AS ORIGIN
       DO 1020 J = 1,4
        XI(J) = XI(J) - XIO
 1020   ETA(J) = ETA(J) - ETAO
C
C  TRANSFORM CENTROID TO REFERENCE COORDINATE SYSTEM
       XCENT = AVX + T1X*XIO + T2X*ETAO
       YCENT = AVY + T1Y*XIO + T2Y*ETAO
       ZCENT = AVZ + T1Z*XIO + T2Z*ETAO
C
C  CONSTANTS
       XI3M1 = XI(3) - XI(1)
       ETA2M4 = ETA(2) - ETA(4)
C
C  COMPUTE AREA AND VOLUME OF ELEMENTS
       AREA = 0.5 * XI3M1 * ETA2M4
      AREAT = AREAT + AREA
       DELVOL = AREA * NY * YCENT
      VOL = VOL + DELVOL
      L = L + 1
      IF (PRINTS.EQ.0) GO TO 1770
C  PRINT RESULTS OF CALCULATIONS TO DETERMINE ELEMENT CHARACTERISTICS
 1700 IF (NPRT.GE.9) GO TO 1750
       NPRT = NPRT + 1
      IF (I .EQ. 1) GO TO 1760
       WRITE (6,4005) I, XIN, NX, XCENT, AREA,L,YIN,NY,YCENT,DELVOL,ZIN,
```

DECK GRPB

```
     1  NZ,ZCENT,VOL                                                     GRPB
      GO TO 1770                                                         GRPB
 1750 NPRT = 0                                                           GRPB
      CALL HEADR2                                                        GRPB
      WRITE (6,4002)                                                     GRPB
 1760 WRITE (6, 4010) N, I, XIN, NX, XCENT, AREA,L,YIN,NY,YCENT,DELVOL,  GRPB
     1 ZIN,NZ,ZCENT,VOL                                                  GRPB
 1770 IF (AREA .LT. 0.1E-09) GO TO 2000                                  GRPB
C                                                                        GRPB
C  CHECK IF NEW GRID IS REQUIRED AND PREPARE GRID                        GRPB
      IF (IFADV .EQ. 0) GO TO 471                                        GRPB
C                                                                        GRPB
      IF (NOSCAL .EQ. 0) GO TO 505                                       GRPB
       CALL STOPTV                                                       GRPB
       CALL BRITEV                                                       GRPB
       CALL XSCALV (XLG,XRG,24,0)                                        GRPB
       CALL YSCALV (YBG,YTG,0,24)                                        GRPB
       GO TO 511                                                         GRPB
 505  CALL GRIDIV (2,XLG,XRG,YBG,YTG,DXG,DYG,NG,MG,IG,JG,NXG,NYG)        GRPB
      DO 510 II=1,3                                                      GRPB
 510  CALL APRNTV (0,-12,30,VTITLE,8,689)                                GRPB
      DO 520 II=1,3                                                      GRPB
 520  CALL PRINTV (29,HTITLE,391,8)                                      GRPB
 511  IF (IFRAME.EQ.1 .AND. ISTART.EQ.1) GO TO 521                       GRPB
      READ (5,522) (HLABEL(II),II=1,15),TYPE                             GRPB
 522  FORMAT(14A4,1A3,11XI2)                                             GRPB
      IF (TYPE .NE. 37) GO TO 300                                        GRPB
      ISTART = 1                                                         GRPB
 521  DO 323 II=1,3                                                      GRPB
      CALL PRINTV (-45,45HHYPERSONIC ARBITRARY-BODY AERODYNAMIC PROGRAM, GRPB
     1  330,1023)                                                        GRPB
 523  CALL PRINTV (59,HLABEL,249,1007)                                   GRPB
      IF (IFRAME .NE. 1) GO TO 525                                       GRPB
      CALL SCSETV (4)                                                    GRPB
      WRITE (16,524) XIN(1),XIN(4)                                       GRPB
 524  FORMAT (1H ,10X11HSTATIONS  =F9.3,8H   AND =F9.3 )                 GRPB
```

```
DECK GRPB

 525  IFADV = 0                                                GRPB  3240
C                                                              GRPB  3250
 471  NXO = THIRD(NX,NY,NZ,PSI,THETA,PHI)                      GRPB  3260
      IF (NXO.LE.0.0 .AND. ISHAD.EQ.0) GO TO 571               GRPB  3270
C                                                              GRPB  3280
C                                                              GRPB  3290
C  CALCULATE POINTS TO BE PLOTTED                              GRPB  3300
 530  YO1 = FIRST(XIN(1),YIN(1),ZIN(1),A1,A2,A3)               GRPB  3310
      YO2 = FIRST(XIN(2),YIN(2),ZIN(2),A1,A2,A3)               GRPB  3320
      YO3 = FIRST(XIN(3),YIN(3),ZIN(3),A1,A2,A3)               GRPB  3330
      YO4 = FIRST(XIN(4),YIN(4),ZIN(4),A1,A2,A3)               GRPB  3340
      ZO1 = FIRST(XIN(1),YIN(1),ZIN(1),A4,A5,A6)               GRPB  3350
      ZO2 = FIRST(XIN(2),YIN(2),ZIN(2),A4,A5,A6)               GRPB  3360
      ZO3 = FIRST(XIN(3),YIN(3),ZIN(3),A4,A5,A6)               GRPB  3370
      ZO4 = FIRST(XIN(4),YIN(4),ZIN(4),A4,A5,A6)               GRPB  3380
C                                                              GRPB  3390
      YIN2(1) = YO1                                            GRPB  3400
      YIN2(2) = YO2                                            GRPB  3410
      YIN2(3) = YO3                                            GRPB  3420
      YIN2(4) = YO4                                            GRPB  3430
      ZIN2(1) = ZO1                                            GRPB  3440
      ZIN2(2) = ZO2                                            GRPB  3450
      ZIN2(3) = ZO3                                            GRPB  3460
      ZIN2(4) = ZO4                                            GRPB  3470
C                                                              GRPB  3480
      CALL APLOTV (4,YIN2,ZIN2,1,1,1,MARKPT,IERR)              GRPB  3490
C                                                              GRPB  3500
      IF (ICS .EQ. 3) GO TO 571                                GRPB  3510
C                                                              GRPB  3520
      IF (ICS.EQ.0 .OR. ICS.EQ.1) GO TO 540                    GRPB  3530
       GO TO 541                                               GRPB  3540
 540   IX1 = NXV(YO1)                                          GRPB  3550
       IY1 = NYV(ZO1)                                          GRPB  3560
      CALL SCERRV (KX,KY)                                      GRPB  3570
       IX2 = NXV(YO2)                                          GRPB  3580
       IY2 = NYV(ZO2)                                          GRPB  3590
```

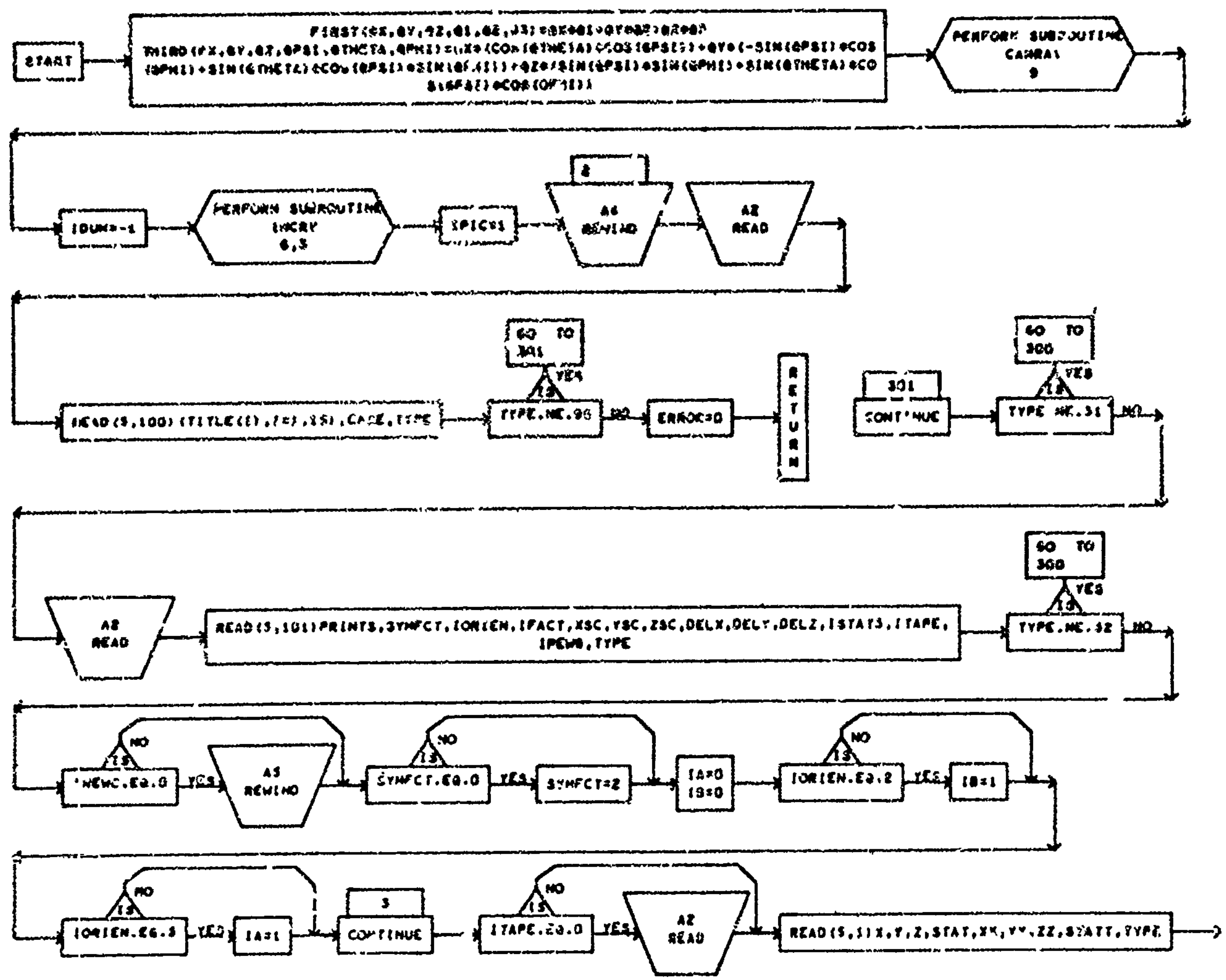
```
DECK GRPB
      CALL SCERRV (KX1,KY1)
      KKXY = KX+KY+KX1+KY1
      IF (KKXY.NE.0) GO TO 541
      IF (NXO.GT.0.0) CALL LINEV (IX1,IY1,IX2,IY2)
      IF (NXO.LE.0.0) CALL DOTLNV(IX1,IY1,IX2,IY2)
C
  541 IF (ICS.EQ.0 .OR. ICS.EQ.2 .OR. ICS.EQ.4) GO TO 550
        GO TO 551
  550   IX1 = NXV(YO2)
        IY1 = NYV(ZO2)
      CALL SCERRV (KX,KY)
        IX2 = NXV(YO3)
        IY2 = NYV(ZO3)
      CALL SCERRV (KX1,KY1)
      KKXY = KX+KY+KX1+KY1
      IF (KKXY.NE.0) GO TO 551
      IF (NXO.GT.0.0) CALL LINEV (IX1,IY1,IX2,IY2)
      IF (NXO.LE.0.0) CALL DOTLNV(IX1,IY1,IX2,IY2)
C
  551 IF (ICS.EQ.0 .OR. ICS.EQ.1 .OR. ICS.EQ.4) GO TO 560
        GO TO 561
  560   IX1 = NXV(YO3)
        IY1 = NYV(ZO3)
      CALL SCERRV (KX,KY)
        IX2 = NXV(YO4)
        IY2 = NYV(ZO4)
      CALL SCERRV (KX1,KY1)
      KKXY = KX+KY+KX1+KY1
      IF (KKXY.NE.0) GO TO 561
      IF (NXO.GT.0.0) CALL LINEV (IX1,IY1,IX2,IY2)
      IF (NXO.LE.0.0) CALL DOTLNV(IX1,IY1,IX2,IY2)
C
  561 IF (ICS.EQ.0 .OR. ICS.EQ.2) GO TO 570
        GO TO 571
  570   IX1 = NXV(YO1)
        IY1 = NYV(ZO1)
```

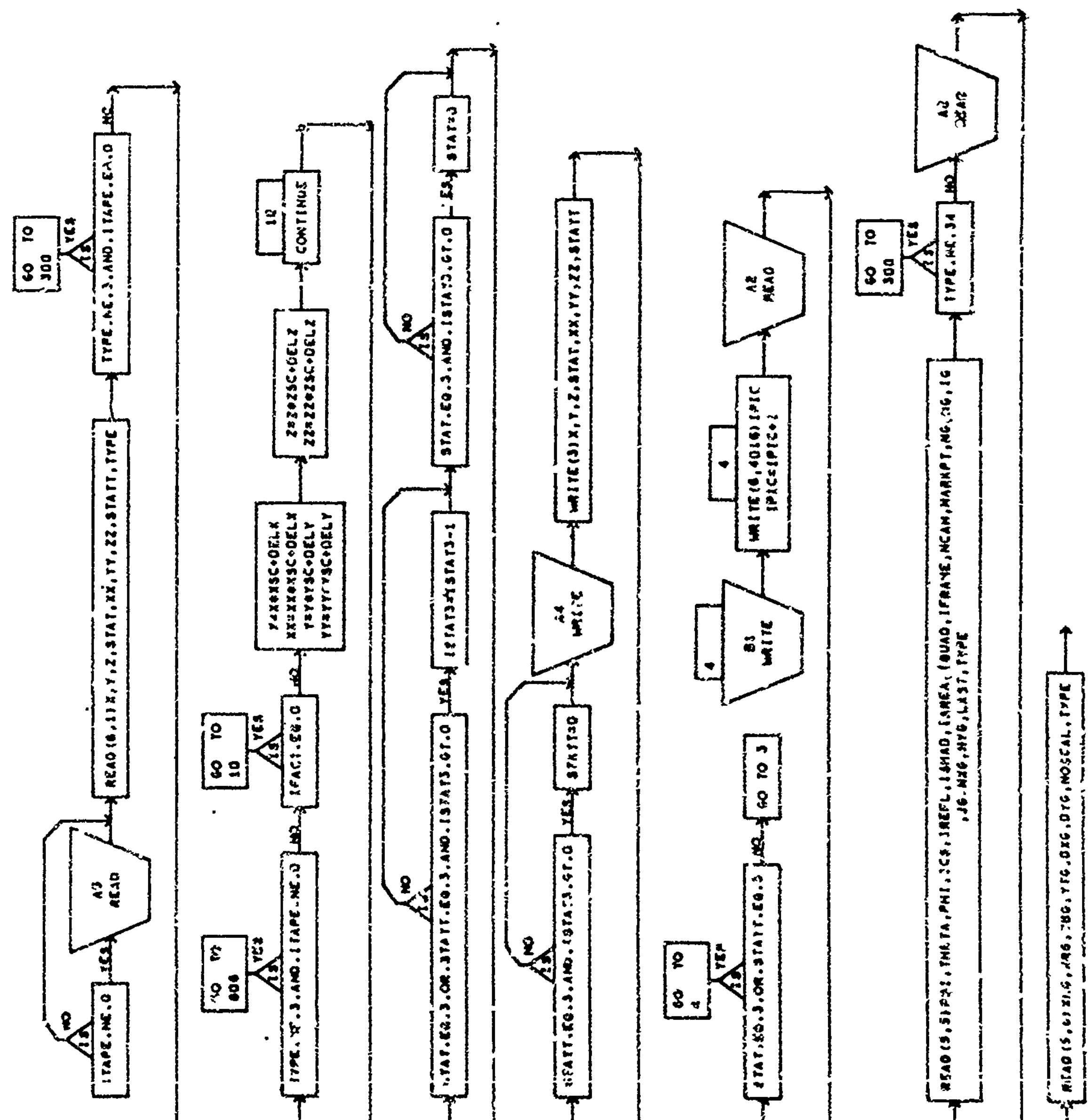
```
DECK GRPB

      CALL SCERRV (KX,KY)                                               GRPB 3960
       IX2 = NXV(YO4)                                                   GRPB 3970
       IY2 = NYV(ZO4)                                                   GRPB 3980
      CALL SCERRV (KX1,KY1)                                             GRPB 3990
      KKXY = KX+KY+KX1+KY1                                              GRPB 4000
      IF (KKXY.NE.0) GO TO 571                                          GRPB 4010
      IF (NXO.GT.0.0) CALL LINEV (IX1,IY1,IX2,IY2)                      GRPB 4020
      IF (NXO.LE.0.0) CALL DOTLNV(IX1 IY1,IX2,IY2)                      GRPB 4030
C                                                                       GRPB 4040
 571  IF (IREFL .EQ. 0 .OR. IRFLG .EQ. 3) GO TO 2000                    GRPB 4050
      IF (IREFL .EQ. 2 .AND. IRFLG .EQ. 1) GO TO 600                    GRPB 4060
      IF (IREFL .EQ. 2 .AND. IRFLG .EQ. 2) GO TO 602                    GRPB 4070
C                                                                       GRPB 4080
C     REFLECT QUADRANT I ELEMENTS TO QUADRANT II                        GRPB 4090
      DO 580 II = 1,4                                                   GRPB 4100
 580   YIN(II) = -YIN(II)                                               GRPB 4110
       NY = -NY                                                         GRPB 4120
       GO TO 604                                                        GRPB 4130
C                                                                       GRPB 4140
C     REFLECT QUADRANT II ELEMENTS TO QUADRANT IV                       GRPB 4150
 600  DO 601 II = 1,4                                                   GRPB 4160
       YIN(II) = -YIN(II)                                               GRPB 4170
 601   ZIN(II) = -ZIN(II)                                               GRPB 4180
       NY = -NY                                                         GRPB 4190
       NZ = -NZ                                                         GRPB 4200
       GO TO 604                                                        GRPB 4210
C                                                                       GRPB 4220
C     REFLECT QUADRANT IV ELEMENTS TO QUADRANT III                      GRPB 4230
 602  DO 603 II = 1,4                                                   GRPB 4240
 603   YIN(II) = -YIN(II)                                               GRPB 4250
       NY = -NY                                                         GRPB 4260
C                                                                       GRPB 4270
C                                                                       GRPB 4280
 604  IRFLG = IRFLG + 1                                                 GRPB 4290
      IF (IREFL .EQ. 1) IRFLG = 3                                       GRPB 4300
      GO TO 471                                                         GRPB 4310
```

```
DECK GRPB

C
C
 2000 CONTINUE
 2001 IF (STAT .LT. 2) GO TO 480
       NPRT = NPRT + 1
      WRITE (6,472) AREAT,L,VOL
      NN = NN - 1
      N = - 1
      IF (IAREA .EQ. 0) GO TO 475
       CALL SCSETV (3)
       WRITE (16,472) AREAT,L
 472   FORMAT(1H ,25X31HTOTAL AREA OF INPUT ELEMENTS  = F14.4,
     1       6X26HTOTAL NUMBER OF ELEMENTS = I5/1H ,23X,
     2 33H TOTAL VOLUME OF INPUT ELEMENTS =F12.3)
 475  IF (IFRAME .EQ. 2) IFADV = 1
 480  IF (IFRAME .EQ. 1) IFADV = 1
 485  IF (IFADV .EQ. 1) CALL FRAMEV (0)
C
C
C  TEST FOR END OF CASE
 2020 IF (STAT .NE. 3) GO TO 80
      IF (LAST .EQ. 1) GO TO 2
      PRINTS=0
      GO TO 4
C
C
C  ERROR CHECK ON READING CARDS
 300  WRITE (6,4003)
C
 4003 FORMAT (1H0,47H*****YOU HAVE MADE AN ERROR EITHER IN CARD TYPE
     1 49H INDICATION OR CARD ORDER - CHECK YOUR CARDS*****  )
      READ (5,810) (CARD(II),II=1,20)
 810  FORMAT (20A4)
      WRITE (6,805) (CARD(II),II=1,20)
 805  FORMAT (1H0,45H THE CARD LOCATED JUST BEFORE THE CARD LISTED
     1 18H BELOW IS IN ERROR,/1H ,10X,20A4)
```

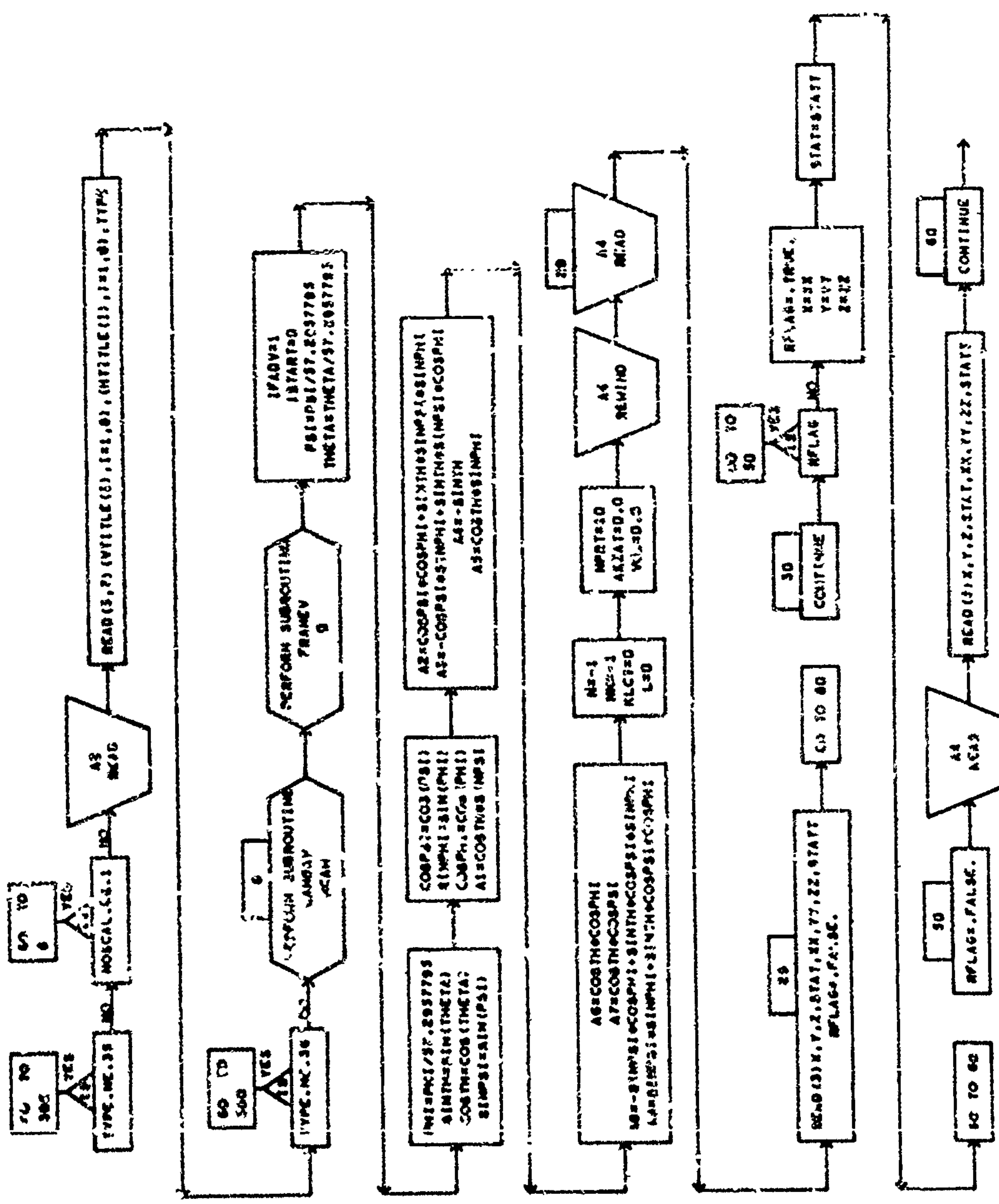
```
DECK GRPB

      GO TO 807
 306  BACKSPACE 8
      READ (8,810) (CARD(II),II=1,20)
      WRITE (6,808) (CARD(II),II=1,20)
 8C8  FORMAT (1H0,46H***THE FOLLOWING CARD ON TAPE 8  S IN ERROR***,/
     1 1H ,10X,20A4)
 807  CALL FRAMEV (0)
      CALL SCSETV (4)
      WRITE (16,4004)
 4004 FORMAT (1H ,45HNO MORE SC-4020 DATA IS PLOTTED BECAUSE OF AN
     126H ERROR IN YOUR INPUT CARDS )
      ERROR = 1
      RETURN
C
 4002 FORMAT (1H0,28H  INPUT SURFACE ELEMENT DATA/1H0,6X1HN3X1HMZ(1HX,
     1 3(13X,1HX),11X2HNX9X5HXCENT9X4HAREA8X1HL ,/1H ,5X, 4(13X,1HY),
     2 11X2HNY9X5HYCENT ,7X,7HDELTA V,/1H ,5X,4(13X,1HZ),11X2HNZ,
     3 9X,5HZCENT ,7X,6HVOLUME,/1H  )
 4005 FORMAT (1H0,7X, I4, 1P4E14.5,0PF10.6,1P2E14.5,I6,2(/12X,4E14.5,
     1 0PF10.6,1P2E14.5)  )
 4010 FORMAT (1H0,3X, 2I4,1P4E14.5,0PF10.6,1P2E14.5,I6,2(/12X,4E14.5,
     1 0PF10.6,1P2E14.5)  )
C
      END
```

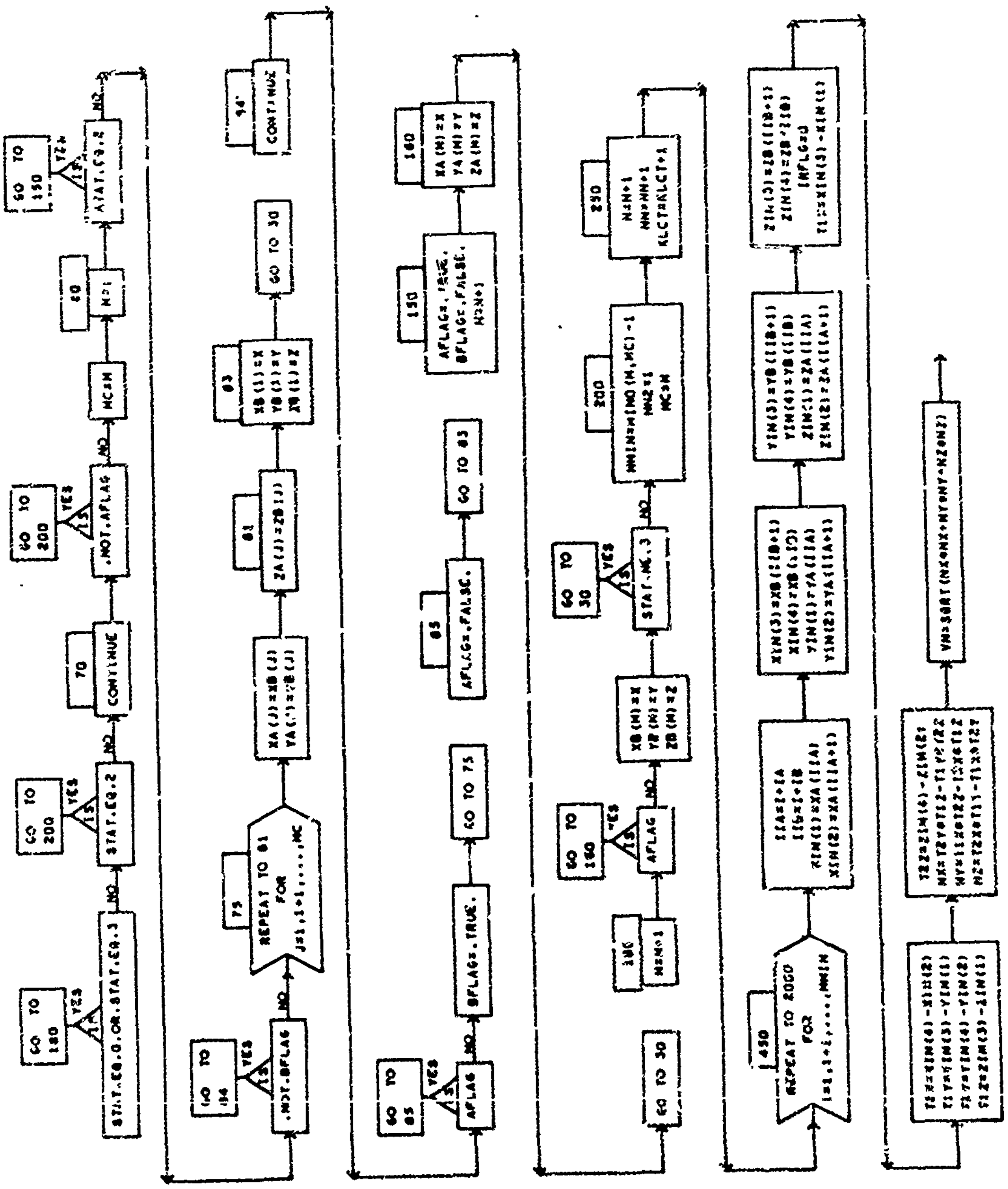
SUBROUTINE PICTUR

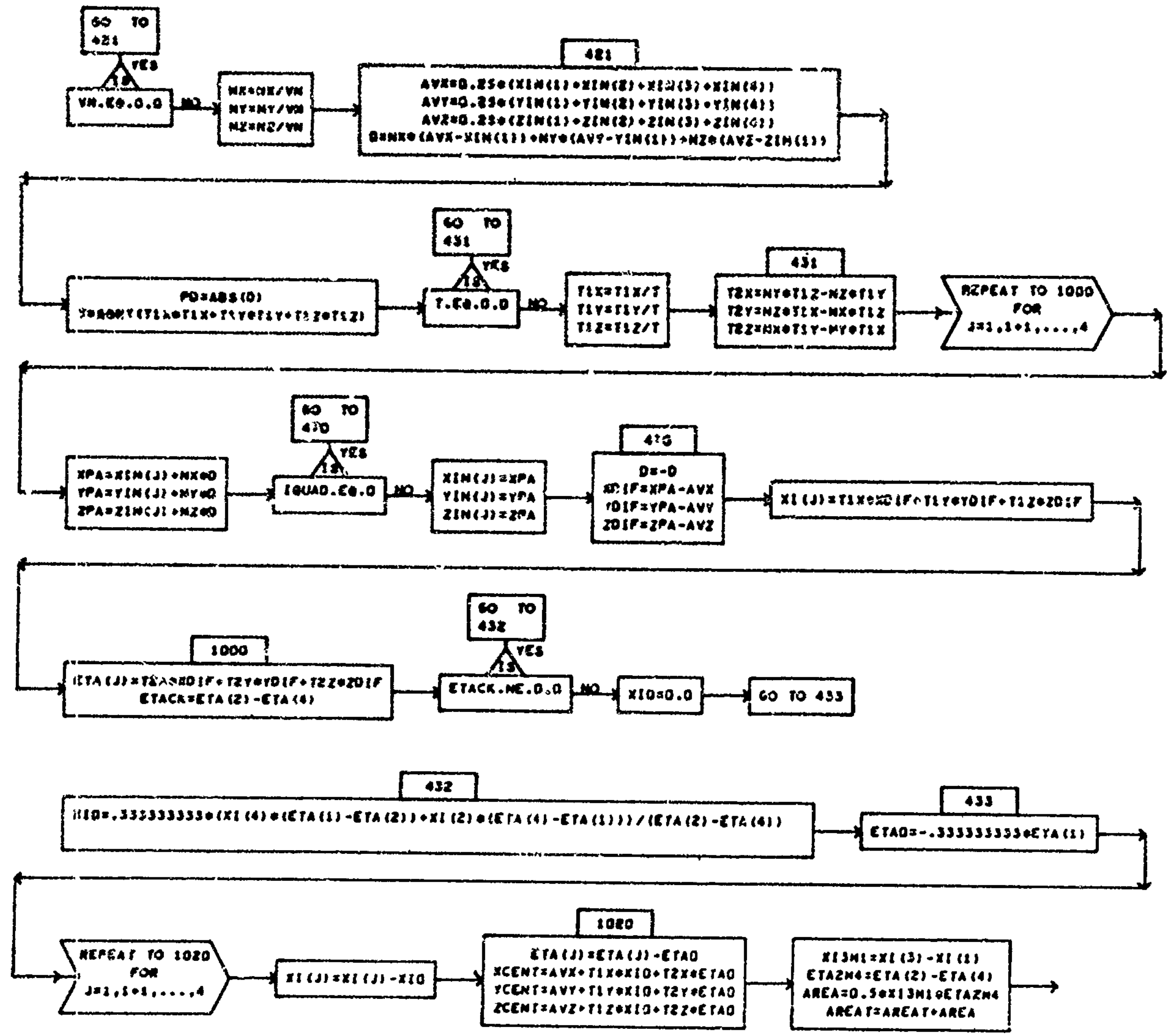


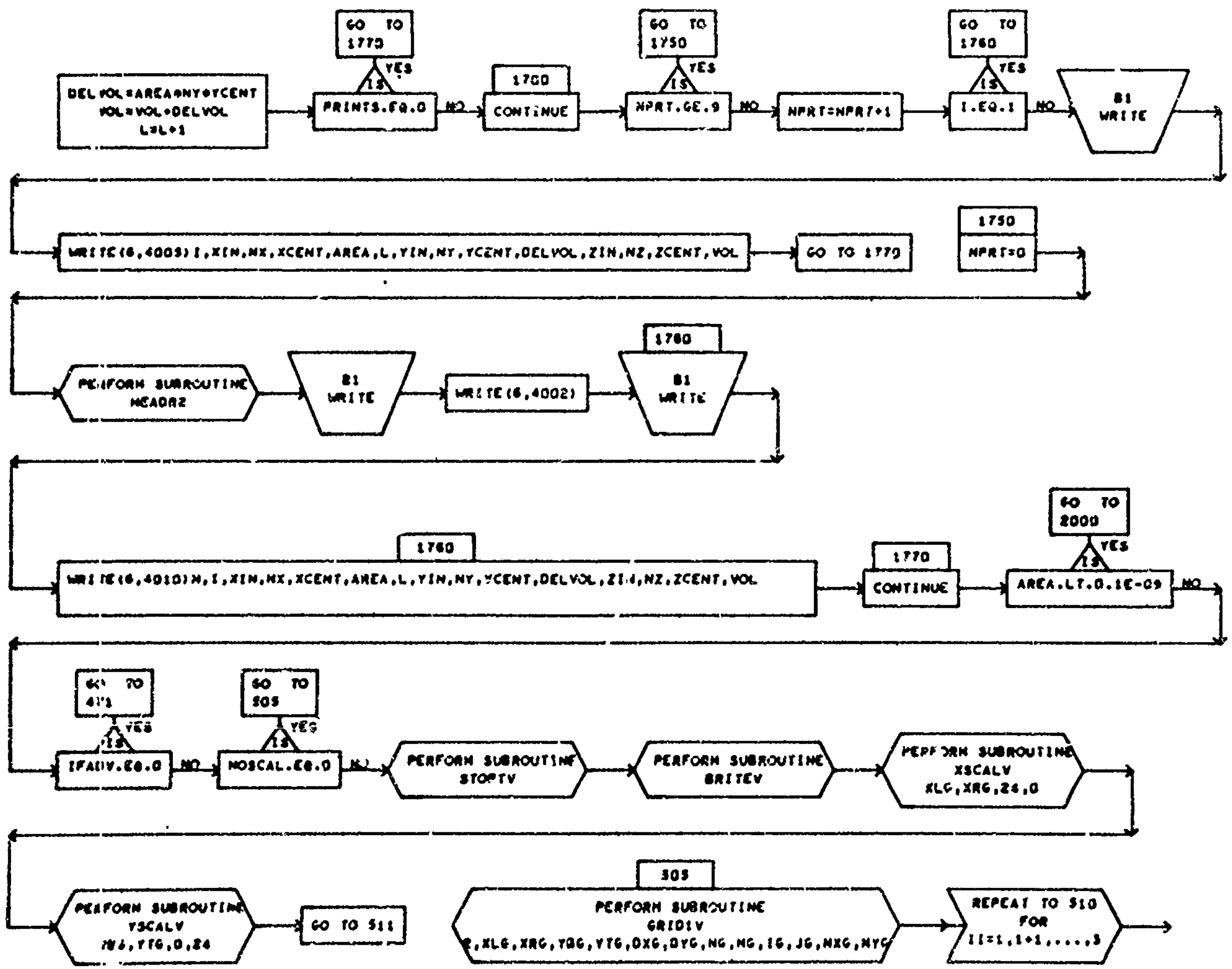


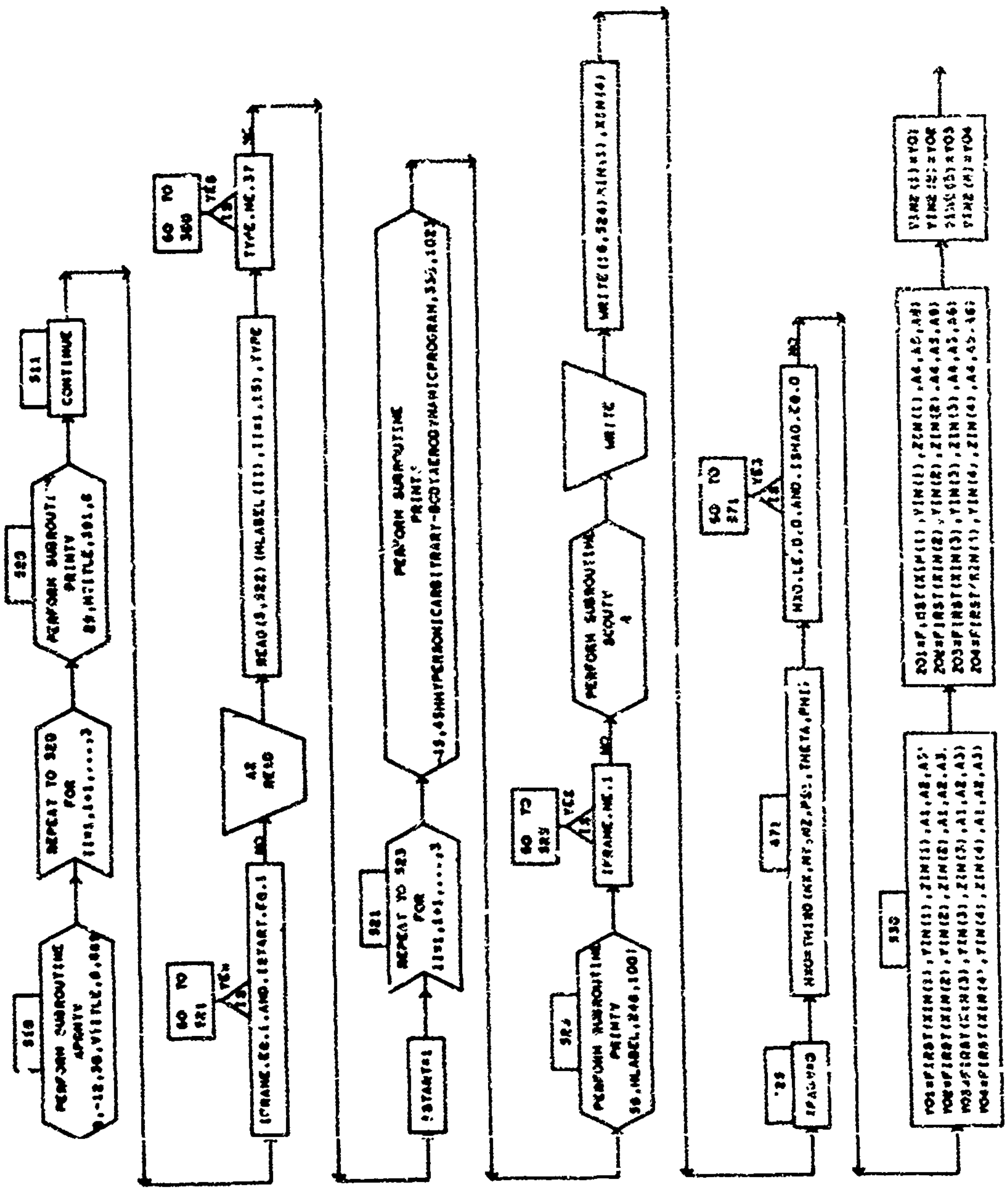
ITAPE.NE.0
IS
NO
YES
A3
READ
READ(8,1)X,Y,Z,STAT,XX,YY,ZZ,STATT,TYPE
TYPE.NE.3.AND.ITAPE.EQ.0
GO TO
300
IFACT.EQ.0
GO TO
10
Z=Z*ZSC+DELZ
ZZ=ZZ*ZSC+DELZ
12
CONTINUE
ISTAT3=ISTAT3-1
STAT.EQ.3.AND.ISTAT3.GT.0
STATT=0
A4
WRITE
GO TO
4
GO TO 3
4
B1
WRITE
A2
READ
TYPE.NE.34

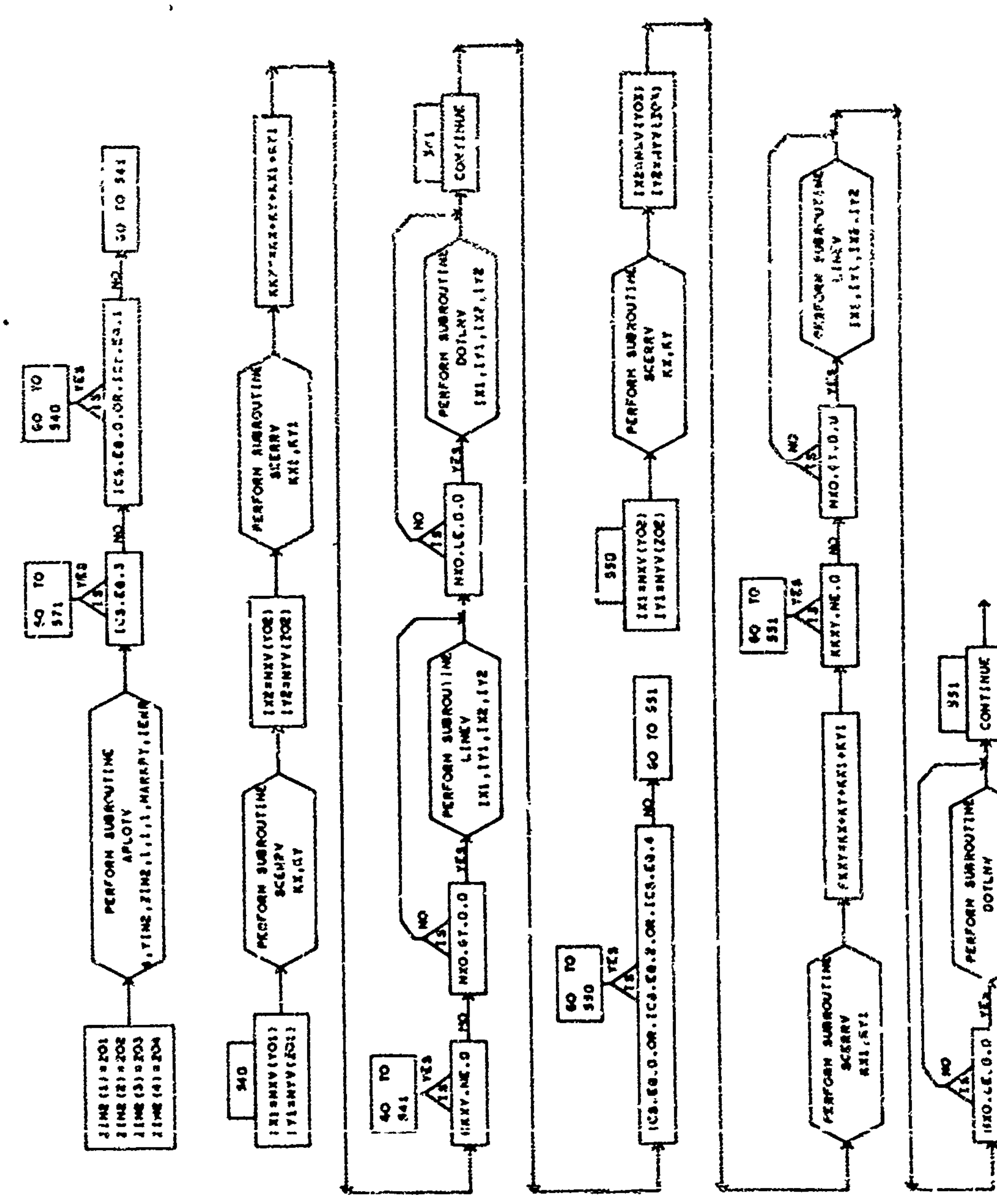


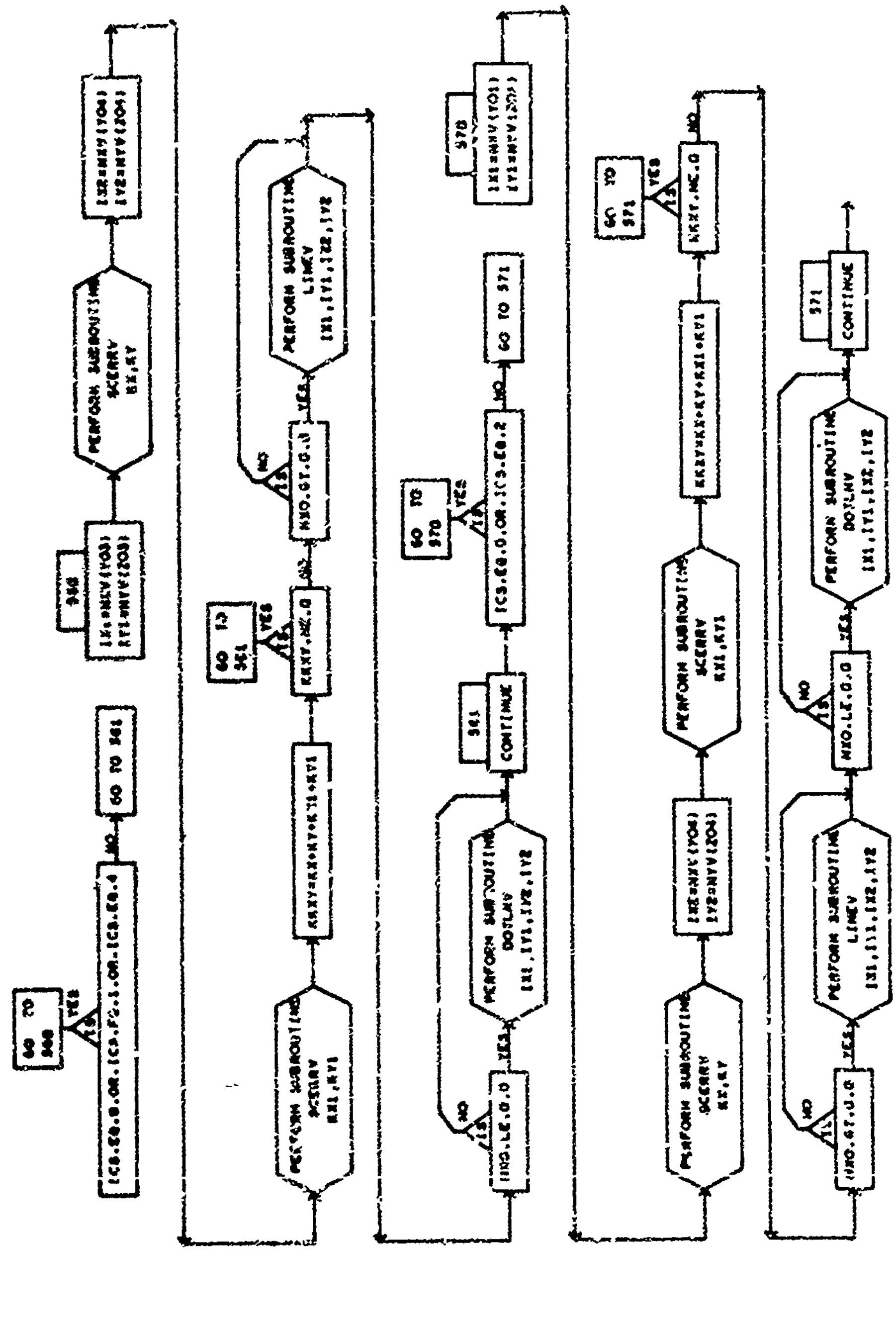


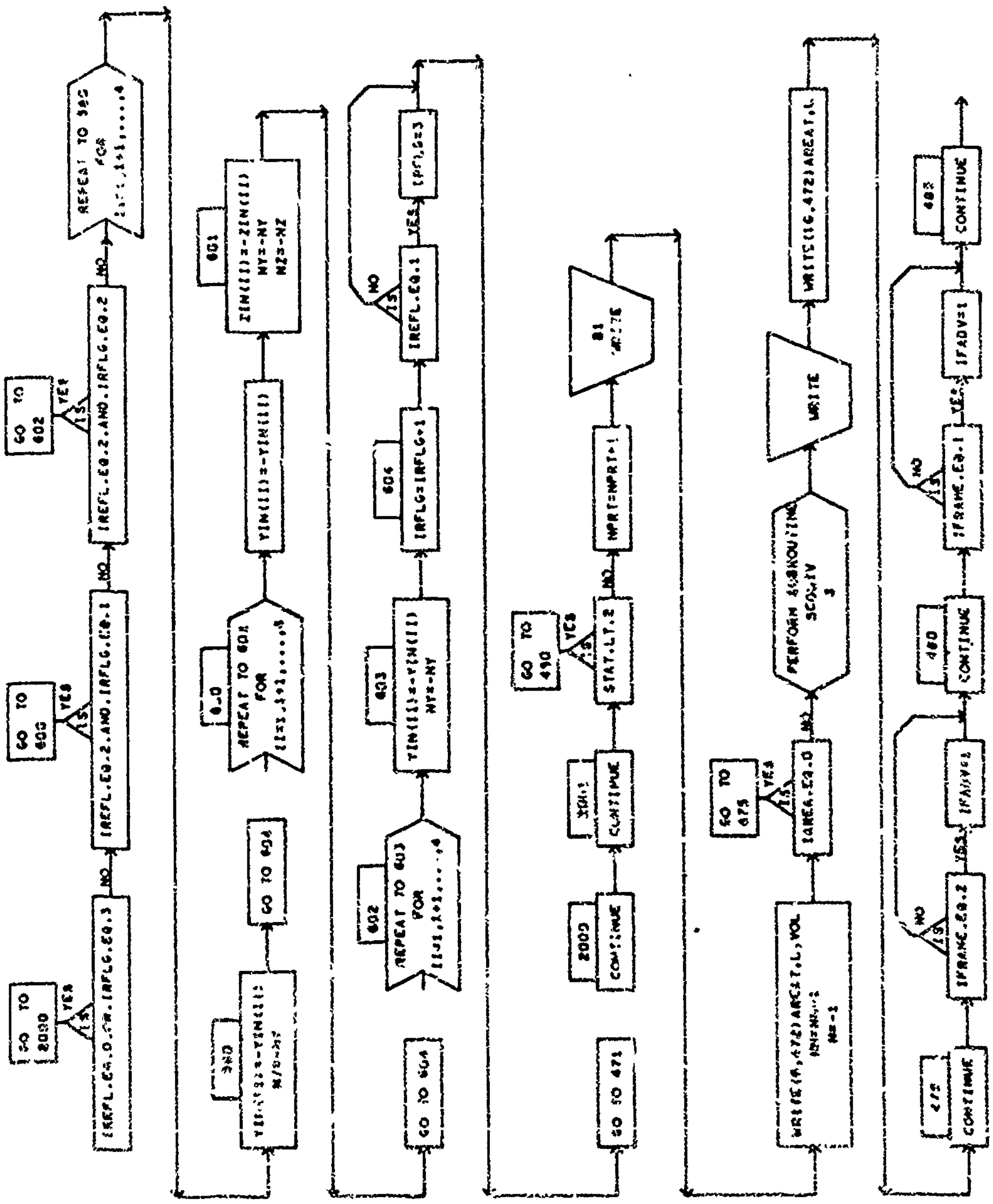

GO TO 421
YES
VN.EQ.0.0
NO
NX=NX/VN
NY=NY/VN
NZ=NZ/VN
421
AVX=0.25*(XIN(1)+XIN(2)+XIN(3)+XIN(4))
AVY=0.25*(YIN(1)+YIN(2)+YIN(3)+YIN(4))
AVZ=0.25*(ZIN(1)+ZIN(2)+ZIN(3)+ZIN(4))
D=NX*(AVX-XIN(1))+NY*(AVY-YIN(1))+NZ*(AVZ-ZIN(1))
PD=ABS(D)
GO TO 431
YES
T.EQ.0.0
NO
T1X=T1X/T
T1Y=T1Y/T
T1Z=T1Z/T
431
T2X=NY*T1Z-NZ*T1Y
T2Y=NZ*T1X-NX*T1Z
T2Z=NX*T1Y-NY*T1X
REPEAT TO 1000
FOR
J=1,1+1,....,4
GO TO 470
YES
IQUAD.EQ.0
NO
XIN(J)=XPA
YIN(J)=YPA
ZIN(J)=ZPA
470
D=-D
XDIF=XPA-AVX
YDIF=YPA-AVY
ZDIF=ZPA-AVZ
1000
ETACK=ETA(2)-ETA(4)
GO TO 432
YES
ETACK.NE.0.0
NO
XI0=0.0
GO TO 433
432
433
ETA0=-.333333333*ETA(1)
REPEAT TO 1020
FOR
J=1,1+1,....,4
XI(J)=XI(J)-XI0
1020
ETA(J)=ETA(J)-ETA0
XCENT=AVX+T1X*XI0+T2X*ETA0
YCENT=AVY+T1Y*XI0+T2Y*ETA0
ZCENT=AVZ+T1Z*XI0+T2Z*ETA0
ETA2M4=ETA(2)-ETA(4)
AREAT=AREAT+AREA

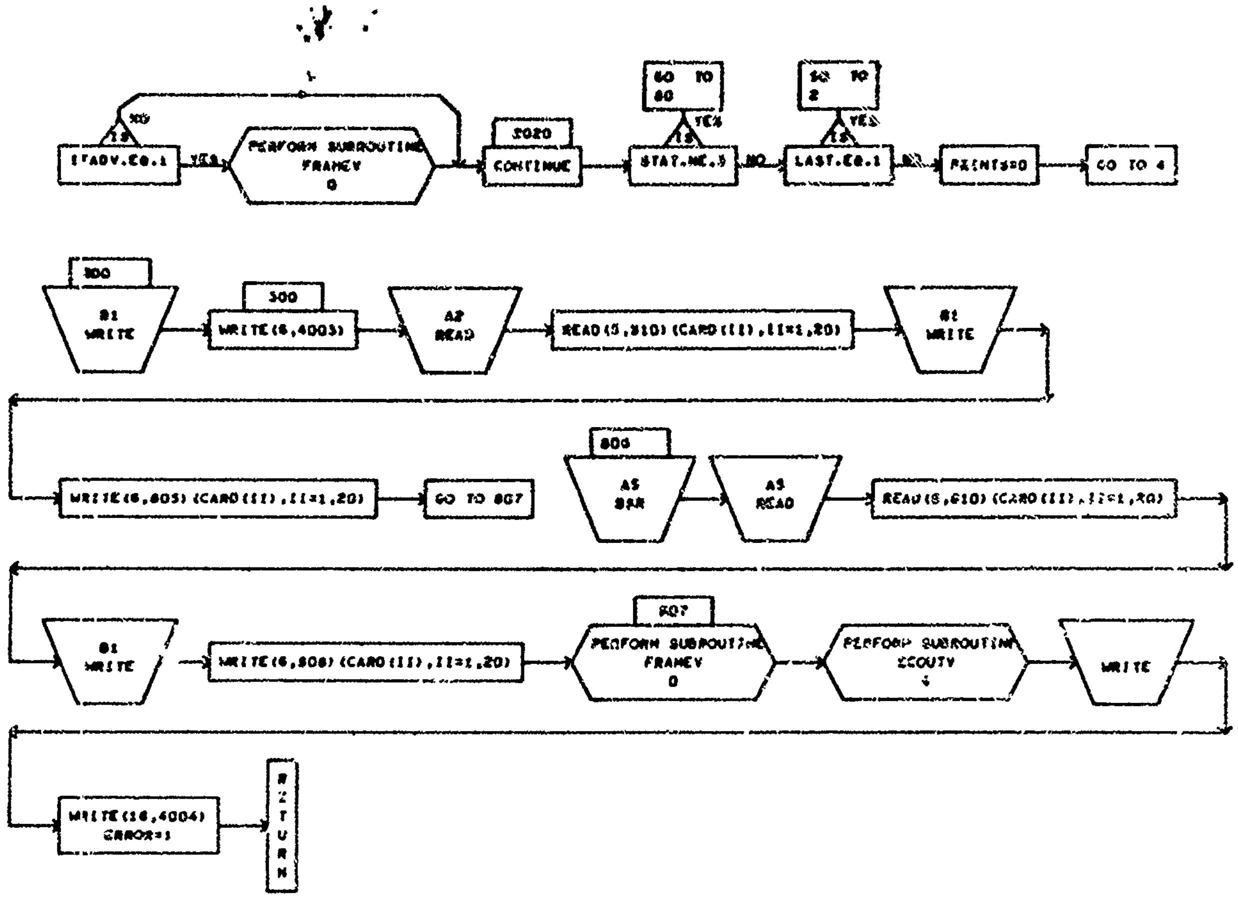

DELVOL=AREA*NY*YCENT
VOL=VOL+DELVOL
L=L+1
GO TO 1770
YES
IS
PRINTS.EQ.0
NO
1700
CONTINUE
GO TO 1750
YES
IS
NPRT.GE.9
NO
NPRT=NPRT+1
GO TO 1760
YES
IS
I.EQ.1
NO
B1
WRITE
WRITE(6,4005)I,XIN,NX,XCENT,AREA,L,YIN,NY,YCENT,DELVOL,ZIN,NZ,ZCENT,VOL
GO TO 1770
1750
NPRT=0
PERFORM SUBROUTINE
HEADR2
B1
WRITE
WRITE(6,4002)
1760
B1
WRITE
1760
WRITE(6,4010)N,I,XIN,NX,XCENT,AREA,L,YIN,NY,YCENT,DELVOL,ZIN,NZ,ZCENT,VOL
1770
CONTINUE
GO TO 2000
YES
IS
AREA.LT.0.1E-09
NO
GO TO 411
YES
IS
IFAUV.EQ.0
NO
GO TO 505
YES
IS
NOSCAL.EQ.0
NO
PERFORM SUBROUTINE
STOPTV
PERFORM SUBROUTINE
BRITEV
PERFORM SUBROUTINE
XSCALV
XLG,XRG,24,0
PERFORM SUBROUTINE
YSCALV
YBG,YTG,0,24
GO TO 511
505
PERFORM SUBROUTINE
GRIDIV
2,XLG,XRG,YBG,YTG,DXG,DYG,NG,NG,IG,JG,NXG,NYG
REPEAT TO 510
FOR
II=1,1+1,...,3




PERFORM SUBROUTINE
APLOTV
GO TO 571
GO TO 540
GO TO 541
540
PERFORM SUBROUTINE
SCERRV
KX,KY
PERFORM SUBROUTINE
LINEV
PERFORM SUBROUTINE
DOTLNV
CONTINUE
541
GO TO 550
GO TO 551
550
GO TO 551
CONTINUE
551
YES
NO
IS




REPEAT TO 595 FOR
IREFL.EQ.2.AND.IRFLG.EQ.2
IREFL.EQ.2.AND.IRFLG.EQ.1
GO TO 602
GO TO 2030
YES
NO
IS
REPEAT TO 603 FOR
YIN(I)=-YIN(I)
GO TO 604
601
603
604
602
IRFLG=IRFLG+1
IREFL.EQ.1
GO TO 471
2000
CONTINUE
NPRT=NPRT+1
STAT.LT.2
GO TO 490
WRITE
PERFORM SUBROUTINE
IAREA.EQ.0
GO TO 475
IFRAME.EQ.2
IFRAME.EQ.1
IFADV=1
CONTINUE
475
480
483



IFADV.EQ.1
IS
NO
YES
PERFORM SUBROUTINE FRAMEV 0
3020
CONTINUE
STAT.NE.3
IS
YES
GO TO 80
NO
LAST.EQ.1
IS
YES
GO TO 2
NO
PRINTS=0
GO TO 4
300
B1 WRITE
300
WRITE(6,4003)
A2 READ
READ(5,810) (CARD(II),II=1,20)
B1 WRITE
WRITE(6,805) (CARD(II),II=1,20)
GO TO 807
806
A5 BAR
A5 READ
READ(5,610) (CARD(II),II=1,20)
B1 WRITE
WRITE(6,808) (CARD(II),II=1,20)
807
PERFORM SUBROUTINE FRAMEV 0
PERFORM SUBROUTINE ECOUTV 4
WRITE
WRITE(16,4004) ERROR=1
RETURN

SYMBOLS USED IN SUBROUTINE PICTUR

AFLAG	L	U	INPUT DATA READ CONTROL FLAG	PICTUR
AREA	R	U	ELEMENT AREA	PICTUR
AREAT	R	U	TOTAL AREA	PICTUR
AVX	R	U	AVERAGE POINT COORDINATE-X	PICTUR
AVY	R	U	AVERAGE POINT COORDINATE-Y	PICTUR
AVZ	R	U	AVERAGE POINT COORDINATE-Z	PICTUR
A1	R	U	ROTATION MATRIX CONSTANT	PICTUR
A2	R	U	ROTATION MATRIX CONSTANT	PICTUR
A3	R	U	ROTATION MATRIX CONSTANT	PICTUR
A4	R	U	ROTATION MATRIX CONSTANT	PICTUR
A5	R	U	ROTATION MATRIX CONSTANT	PICTUR
A6	R	U	ROTATION MATRIX CONSTANT	PICTUR
A7	R	U	ROTATION MATRIX CONSTANT	PICTUR
A8	R	U	ROTATION MATRIX CONSTANT	PICTUR
A9	R	U	ROTATION MATRIX CONSTANT	PICTUR
BFLAG	L	U	INPUT DATA READ CONTROL FLAG	PICTUR
CARD	R	D	ARRAY FOR READING IN 80 COLUMN CARD	PICTUR
CASE	I	C	CASE NUMBER	PICTUR
COSPHI	R	U	COSINE OF PHI	PICTUR
COSPSI	R	U	COSINE OF PSI	PICTUR
COSTH	R	U	COSINE OF THETA	PICTUR
D	R	U	CORNER POINT PROJECTION DISTANCE	PICTUR
DELVOL	R	U	ELEMENT VOLUME CONTRIBUTION	PICTUR
DELX	R	U	GEOMETRY DATA X-INCREMENT	PICTUR
DELY	R	U	GEOMETRY DATA Y-INCREMENT	PICTUR
DELZ	R	U	GEOMETRY DATA Z-INCREMENT	PICTUR
DXG	R	U	GRID DELTA-X INCREMENT	PICTUR
DYG	R	U	GRID DELTA-Y INCREMENT	PICTUR
ERROR	I	C	ERROR FLAG	PICTUR
ETA	R	D	COORDINATE IN ELEMENT COORDINATE SYSTEM	PICTUR
ETACK	R	U	ETA CHECK PARAMETER	PICTUR
ETAO	R	U	COORDINATE IN ELEMENT COORDINATE SYSTEM	PICTUR
ETA2M4	R	U	CONSTANT IN AREA EQUATION	PICTUR
HLABEL	R	D	HORIZONTAL LABEL	PICTUR
HTITLE	R	D	VERTICAL LABEL	PICTUR
I	I	U	ELEMENT NUMBER IN COLUMN	PICTUR

```
SYMBOLS USED IN SUBROUTINE PICTUR

IA       I  U  FLAG TO CONTROL SHIFTING OF COLUMN DATA POINTS FOR IORIEN=3      PICTUR
IAREA    I  U  SURFACE AREA PRINT FLAG                                          PICTUR
IB       I  U  FLAG TO CONTROL SHIFTING OF COLUMN DATA POINTS FOR IORIEN=2      PICTUR
ICS      I  U  POINT CORRECT FLAG                                               PICTUR
IDUM     I  U  DUMMY VARIABLE                                                   PICTUR
IERR     I  U  ERROR FLAG                                                       PICTUR
IFACT    I  U  SCALE FACTOR FLAG                                                PICTUR
IFADV    I  U  FRAME FLAG                                                       PICTUR
IFRAME   I  U  FRAME ADVANCE FLAG                                               PICTUR
IG       I  U  VERTICAL GRID LINE LABEL CONTROL FLAG                            PICTUR
IIA      I  U  DATA SHIFTING CONTROL PARAMETER (IORIEN=3)                       PICTUR
IIB      I  U  DATA SHIFTING CONTROL PARAMETER (IORIEN=2)                       PICTUR
IORIEN   I  U  ELEMENT ORIENTATION (NOT USED)                                   PICTUR
IPIC     I  U  FRAME NUMBER                                                     PICTUR
IQUAD    I  U  QUADRILATERAL PLOT FLAG                                          PICTUR
IREFL    I  U  REFLECTION ELEMENTS CONTROL FLAG                                 PICTUR
IREWB    I  U  TAPE 8 REWIND FLAG                                               PICTUR
IRFLG    I  U  REFLECTION CONTROL FLAG                                          PICTUR
ISHAD    I  U  SHADOW ELEMENT FLAG                                              PICTUR
ISTART   I  U  CONTROL FLAG                                                     PICTUR
ISTAT3   I  U  NUMBER OF STATUS = 3 POINTS IN DECK                              PICTUR
ITAPE    I  U  GEOMETRY SOURCE FLAG                                             PICTUR
IX1      I  U  X-RASTER COORDINATE OF FIRST POINT                               PICTUR
IX2      I  U  X-RASTER COORDINATE OF SECOND POINT                              PICTUR
IY1      I  U  Y-RASTER COORDINATE OF FIRST POINT                               PICTUR
IY2      I  U  Y-RASTER COORDINATE OF SECOND POINT                              PICTUR
JG       I  U  HORIZONTAL GRID LINE LABEL CONTROL FLAG                          PICTUR
KKXY     I  U  OFF-SCALE DETECTION FLAG                                         PICTUR
KLCT     I  U  COUNTER                                                          PICTUR
KX       I  U  OFF-SCALE DETECTION FLAG                                         PICTUR
KX1      I  U  OFF-SCALE DETECTION FLAG                                         PICTUR
KY       I  U  OFF-SCALE DETECTION FLAG                                         PICTUR
KY1      I  U  OFF-SCALE DETECTION FLAG                                         PICTUR
L        I  U  ELEMENT NUMBER                                                   PICTUR
LAST     I  U  LAST PLOT CONTROL FLAG                                           PICTUR
M        I  U  DATA READ IN CONTROL FLAG                                        PICTUR
```

SYMBOLS USED IN SUBROUTINE PICTUR

MARKPT	I	U	PLOTTING SYMBOL CODE	PICTUR
MC	I	U	DATA READ IN CONTROL NUMBER	PICTUR
MG	I	U	HORIZONTAL LINE EMPHASIZE FLAG	PICTUR
MMIN	I	U	NUMBER OF ELEMENTS IN A COLUMN	PICTUR
N	I	U	COLUMN NUMBER	PICTUR
NCAM	I	U	CAMERA SELECTION FLAG	PICTUR
NG	I	U	VERTICAL LINE EMPHASIZE FLAG	PICTUR
NN	I	U	COLUMN ELEMENT COUNTER	PICTUR
NN2	I	U	COUNTER	PICTUR
NOSCAL	I	U	NO GRID FLAG	PICTUR
NPRT	I	U	LINE COUNTER	PICTUR
NX	R	U	ELEMENT DIRECTION COSINE-X	PICTUR
NXG	I	U	NUMBER OF CHARACTERS IN X-SCALE NUMBER LABELS	PICTUR
NXO	R	U	DIRECTION COSINE OUT OF PLANE OF PAPER	PICTUR
NY	R	U	ELEMENT DIRECTION COSINE-Y	PICTUR
NYG	I	U	NUMBER OF CHARACTERS IN Y-SCALE NUMBER LABELS	PICTUR
NZ	R	U	ELEMENT DIRECTION COSINE-Z	PICTUR
PAGE	I	C	PAGE NUMBER	PICTUR
PD	R	U	CORNER POINT PROJECTION DISTANCE	PICTUR
PHI	R	U	ROLL ANGLE, DEGREES	PICTUR
PRINTS	I	U	ELEMENT DATA PRINT FLAG	PICTUR
PSI	R	U	YAW ANGLE	PICTUR
RFLAG	L	U	INPUT DATA READ CONTROL FLAG	PICTUR
SINPHI	R	U	SIN OF PHI	PICTUR
SINPSI	R	U	SIN OF PSI	PICTUR
SINTH	R	U	SIN OF THETA	PICTUR
STAT	I	U	COORDINATE POINT STATUS FLAG	PICTUR
START	I	U	COORDINATE POINT STATUS FLAG	PICTUR
SYMFCT	I	U	SYMMETRY FLAG	PICTUR
T	R	U	UNIT VECTOR	PICTUR
THETA	R	U	PITCH ANGLE	PICTUR
TITLE	R	C	TITLE	PICTUR
TYPE	I	U	CARD TYPE NUMBER	PICTUR
T1X	R	U	X-COMPONENT OF VECTOR T1	PICTUR
T1Y	R	U	Y-COMPONENT OF VECTOR T1	PICTUR
T1Z	R	U	Z-COMPONENT OF VECTOR T1	PICTUR

SYMBOLS USED IN SUBROUTINE PICTUR

T2X	R	U	X-COMPONENT OF VECTOR T2	PICTUR
T2Y	P	U	Y-COMPONENT OF VECTOR T2	PICTUR
T2Z	R	U	Z-COMPONENT OF VECTOR T2	PICTUR
VN	R	U	VECTOR LENGTH	PICTUR
VOL	R	U	TOTAL VOLUME	PICTUR
VTITLE	R	D	VERTICAL SCALE TITLE	PICTUR
X	R	U	X-COORDINATE	PICTUR
XA	R	D	X-COORDINATE	PICTUR
XB	R	D	X-COORDINATE	PICTUR
XCENT	R	U	ELEMENT CENTROID COORDINATE-X	PICTUR
XDIF	R	U	COORDINATE DIFFERENCE-X	PICTUR
XI	R	D	COORDINATE IN ELEMENT COORDINATE SYSTEM	PICTUR
XIN	R	D	ELEMENT COORDINATES-X	PICTUR
XIO	R	U	CENTROID IN ELEMENT COORDINATE SYSTEM	PICTUR
XI3M1	R	U	CONSTANT FOR AREA EQUATION	PICTUR
XLG	R	U	VALUE OF LEFT SIDE OF HORIZONTAL SCALE	PICTUR
XPA	R	U	COORDINATE OF ELEMENT CORNER POINT	PICTUR
XRG	R	U	VALUE OF RIGHT SIDE OF HORIZONTAL SCALE	PICTUR
XSC	R	U	X SCA'E FACTOR	PICTUR
XX	R	U	X-COORDINATE	PICTUR
Y	R	U	Y-COORDINATE	PICTUR
YA	R	D	Y-COORDINATE	PICTUR
YB	R	D	Y-COORDINATE	PICTUR
YBG	R	U	VALUE OF BOTTOM OF VERTICAL SCALE	PICTUR
YCENT	R	U	ELEMENT CENTROID COORDINATE-Y	PICTUR
YDIF	R	U	COORDINATE DIFFERENCE-Y	PICTUR
YIN	R	D	ELEMENT COORDINATES-Y	PICTUR
YIN2	R	D	Y-COORDINATE FOR PLOT	PICTUR
YO1	R	U	Y-COORDINATE FOR PLOT-POINT 1	PICTUR
YO2	R	U	Y-COORDINATE FOR PLOT-POINT 2	PICTUR
YO3	R	U	Y-COORDINATE FOR PLOT-POINT 3	PICTUR
YO4	R	U	Y-COORDINATE FOR PLOT-POINT 4	PICTUR
YPA	R	U	COORDINATE OF ELEMENT CORNER POINT	PICTUR
YSC	R	U	Y-SCALE FACTOR	PICTUR
YTG	R	U	VALUE OF TOP OF VERTICAL SCALE	PICTUR
YY	R	U	Y-COORDINATE	PICTUR

SYMBOLS USED IN SUBROUTINE PICTUR

Z	R	U	Z-COORDINATE	PICTUR
ZA	R	D	Z-COORDINATE	PICTUR
ZB	R	D	Z-COORDINATE	PICTUR
ZCENT	R	U	ELEMENT CENTROID COORDINATE-Z	PICTUR
ZDIF	R	U	COORDINATE DIFFERENCE-Z	PICTUR
ZIN	R	D	ELEMENT COORDINATES-Z	PICTUR
ZIN2	R	D	Z-COORDINATE FOR PLOT	PICTUR
Z01	R	U	Z-COORDINATE FOR PLOT-POINT 1	PICTUR
Z02	R	U	Z-COORDINATE FOR PLOT-POINT 2	PICTUR
Z03	R	U	Z-COORDINATE FOR PLOT-POINT 3	PICTUR
Z04	R	U	Z-COORDINATE FOR PLOT-POINT 4	PICTUR
ZPA	R	U	COORDINATE OF ELEMENT CORNER POINT	PICTUR
ZSC	R	U	Z-SCALE FACTOR	PICTUR
ZZ	R	U	Z-COORDINATE	PICTUR

30. SUBROUTINE HEADR2 (DECK GRPC)

a. Algorithm

This routine provides the title at the top of each page of the output and advances the page counter. This routine is very similar to the HEADER routine.

b. Input/Output

Program header is printed at top of page on output Tape 6.

c. Error

None

d. Subroutines Required

None

e. Argument List

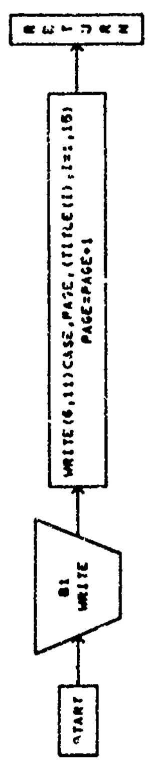
None

f. Length

350 bytes

```
DECK GRPC

      SUBROUTINE HEADR2
C
      DIMENSION  TITLE(15)
C
      COMMON CASE,TITLE,PAGE
C
      INTEGER  PAGE, CASE
C
C  PRINT OUT HEADER AT TOP OF EACH PAGE OF OUTPUT
      WRITE (6,11) CASE,PAGE,(TITLE(I),I=1,15)
   11 FORMAT (1H1,5X,39HQUADRILATERAL CHARACTERISTICS - PICTURE
     1 16H DRAWING PROGRAM ,/1H0,5X,6H  CASE,I5,80X,5HPAGE I4,/
     2 1H0,14A4,A3)
C
C  STEP PAGE NUMBER BY ONE
      PAGE = PAGE + 1
C
      RETURN
      END
```

SUBROUTINE HEADR2



```
SYMBOLS USED IN SUBROUTINE HEADR2

CASE     I  C  CASE NUMBER                                   HEADR2
PAGE     I  C  PAGE NUMBER                                   HEADR2
TITLE    R  C  TITLE                                         HEADR2
```

31. SUBROUTINE PLOT (DECK GRPD)

This routine is used to produce graphically plotted data as obtained from the aerodynamics part of the program.

a. Algorithm

Read in plotter control cards. As directed, read aerodynamic data from Tape 9. Prepare plot scales and grids. Plot data and connect data points as directed.

b. Input/Output

Data Source Control Card (Type 41), Vertical-Title Card (Type 44), Horizontal-Title Card (Type 45), Plotting-Grid Data Card (Type 45), Plot Control Array Card (Type 47), and Horizontal-Lable Card(s) Type 48).
Output plots are on the SC-4020 tape.

c. Error

An error condition occurs when the card type number is wrong.

d. Subroutines Required

None

e. Argument List

None

f. Length

10772 bytes

DECK GRPD

```
                                                                        GRPD 0010
      SUBROUTINE PLOT                                                   GRPD 0020
C      POINT PLOTTER PROGRAM. UP TO 12 ARRAYS CAN BE INPUT              GRPD 0030
C      AND ANY ONE OF THEM PLOTTED AGAINST ANY OTHER                    GRPD 0040
C                                                                       GRPD 0050
      DIMENSION A(20),B(20),C(20),D(20),E(20),F(20),AA(20),BB(20),      GRPD 0060
     1 CC(20),DD(20),EE(20),FF(20),X(20),Y(20),I(100),TITLE(900),W(10), GRPD 0070
     3 PRINT(10),TITLE2(15)                                             GRPD 0080
      COMMON  CASE,TITLE2,PAGE,ERROR                                    GRPD 0090
      INTEGER  ERROR,TYPE,PAGE,CASE                                     GRPD 0100
C                                                                       GRPD 0110
      CALL CAMRAV (9)                                                   GRPD 0120
      REWIND 1                                                          GRPD 0130
      REWIND 10                                                         GRPD 0140
      REWIND 9                                                          GRPD 0150
C     READ FIRST CONTROL CARD                                           GRPD 0160
    1 READ (5,2) NC,IT                                                  GRPD 0170
    2  FORMAT (I4,65XI2)                                                GRPD 0180
C     CHECK TYPE OF CARD                                                GRPD 0190
      IF (IT .EQ. 41) GO TO 5                                           GRPD 0200
C                                                                       GRPD 0210
C     TYPE CARD GOOFY. PRINT ERROR HEADER AND LEAVE                     GRPD 0220
      IT = 41                                                           GRPD 0230
    3 CALL CAMRAV(9)                                                    GRPD 0240
      CALL FRAMEV (0)                                                   GRPD 0250
      CALL SCOUTV                                                       GRPD 0260
      CALL SCSETV (4)                                                   GRPD 0270
      WRITE (16,4) IT,IT                                                GRPD 0280
    4 FORMAT (1H4,15X30HFOR SOME ODD REASON, TYPE CARDI3,1X             GRPD 0290
     1  15HDOES NOT HAVE AI3,31H IN COLUMN 71-72. BETTER LUCK N         GRPD 0300
     2  9HEXT TIME. )                                                   GRPD 0310
      WRITE (6,48) IT,IT                                                GRPD 0320
   48 FORMAT (1H0,15X30HFOR SOME ODD REASON, TYPE CARDI3,1X             GRPD 0330
     1  15HDOES NOT HAVE AI3,31H IN COLUMN 71-72. BETTER LUCK N         GRPD 0340
     2  9HEXT TIME. )                                                   GRPD 0350
      READ (5,49) (TITLE(J),J=1,20)
```

```
DECK GRPD

   49 FORMAT (20A4)                                                    GRPD 0360
      WRITE (6,50) (TITLE(J),J=1,20)                                   GRPD 0370
   50 FORMAT (1H0,25X36HWRITTEN BELOW IS THE IMAGE OF THE CA           GRPD 0380
     1  30HRD FOLLOWING THE INCORRECT ONE//20X,20A4)                   GRPD 0390
      ERROR = 1                                                        GRPD 0400
      GO TO 101                                                        GRPD 0410
C                                                                      GRPD 0420
C     CHECK FOR INPUT TAPE, IF ANY                                     GRPD 0430
    5 I(20) = 0                                                        GRPD 0440
      ICNT  = 0                                                        GRPD 0450
      IF (NC) 28,33,7                                                  GRPD 0460
   33 READ(10)NC,NC,NC,NC,IT                                           GRPD 0470
      WRITE (6,44) NC                                                  GRPD 0480
   44 FORMAT (1H0,31X34HTAPE 10 JUST INSTRUCTED ME TO READI4           GRPD 0490
     1  ,18H CARDS FROM TAPE 9 )                                       GRPD 0500
C                                                                      GRPD 0510
C     CHECK TYPE                                                       GRPD 0520
      IF (IT .EQ. 42) GO TO 43                                         GRPD 0530
C                                                                      GRPD 0540
C     TYPE ERROR                                                       GRPD 0550
      IT = 42                                                          GRPD 0560
      GO TO 3                                                          GRPD 0570
   43 DO 34 J = 1,NC                                                   GRPD 0580
   34  READ (9) A(J),B(J),C(J),D(J),E(J),F(J),AA(J),BB(J),             GRPD 0590
     1  CC(J),DD(J),EE(J),FF(J),IT                                     GRPD 0600
C                                                                      GRPD 0610
C      CHECK TYPE (IT) OF CARD JUST READ                               GRPD 0620
      IF (IT .EQ. 43) GO TO 71                                         GRPD 0630
C                                                                      GRPD 0640
C      TYPE ERROR                                                      GRPD 0650
      IT = 43                                                          GRPD 0660
       GO TO 3                                                         GRPD 0670
   71  CONTINUE                                                        GRPD 0680
       WRITE (6,45) NC                                                 GRPD 0690
   45  FORMAT (1H0,42X11HI JUST READI4,18H CARDS FROM TAPE 9)          GRPD 0700
      IF (I(20) .NE. 0) GO TO 78                                       GRPD 0710
```

DECK GRPD

```
      GO TO 28                                                          GRPD 0720
    7 DO 6 K = 1, NC                                                    GRPD 0730
      DO 6 J = 1, 2                                                     GRPD 0740
      IF (J .EQ. 2) GO TO 69                                            GRPD 0750
      READ(5,8)  A(K),B(K),C(K),D(K),E(K),F(K),AA(K),IT                 GRPD 0760
    8 FORMAT (7F10.0,I2)                                                GRPD 0770
C                                                                       GRPD 0780
C     CHECK TYPE (IT) OF CARD JUST READ                                 GRPD 0790
    9 IF(IT .EQ. 43) GO TO 6                                            GRPD 0800
C                                                                       GRPD 0810
C     TYPE SCREWED UP. ERROR                                            GRPD 0820
      IT = 43                                                           GRPD 0830
      GO TO 3                                                           GRPD 0840
   69 READ(5,70)  BB(K),CC(K),DD(K),EE(K),FF(K),IT                      GRPD 0850
   70 FORMAT(5F10.0,20XI2)                                              GRPD 0860
      GO TO 9                                                           GRPD 0870
    6 CONTINUE                                                          GRPD 0880
      WRITE (6,46) NC                                                   GRPD 0890
   46 FORMAT (1H0,24X33HTAPE 5 JUST INSTRUCTED ME TO READI4,            GRPD 0900
     1   34H RECORDS FROM TAPE 5, WHICH I DID. )                        GRPD 0910
     GO TO 28                                                           GRPD 0920
C                                                                       GRPD 0930
C    READ STUFF FROM TAPE 1                                             GRPD 0940
   51 READ (1) NC,NC,NC,NC,NC                                           GRPD 0950
     DO 52 J = 1,NC                                                     GRPD 0960
   52 READ (1) A(J),B(J),C(J),D(J),E(J),F(J),AA(J),                     GRPD 0970
    1 BB(J),CC(J),DD(J),EE(J),FF(J)                                     GRPD 0980
     REWIND 1                                                           GRPD 0990
     WRITE (6,53) NC                                                    GRPD 1000
   53 FORMAT (1H0,24X33HTAPE 1 JUST INSTRUCTED ME TO READI4,            GRPD 1010
    1  32H CARDS FROM TAPE 1, WHICH I DID. )                            GRPD 1020
C                                                                       GRPD 1030
C    READ IN PLOTTING INFORMATION CARDS                                 GRPD 1040
   28 READ (5,10) (TITLE(J), J=1, 18), IT                               GRPD 10
   10 FORMAT (17A4,A2,I2)                                               GRPD 1060
      IF (IT .EQ. 44) GO TO 72                                          GRPD 1070
```

```
DECK GRPD
      IT = 44
      GO TO 3
   72 READ (5,10) (TITLE(J), J=19,36), IT
      IF (IT .EQ. 45) GO TO 74
      IT = 45
      GO TO 3
   74 READ (5,75) (W(J), J=1, 6), IT
   75 FORMAT (6F10.0,10XI2)
      IF (IT .EQ. 46) GO TO 76
      IT =6
      GO TO 3
   76 READ (5,77) (I(J), J=1,21), IT
   77 FORMAT (3I2,I4,I3,I4,I3,I5,I4,I3,I4,I3,I2,I3,I2,I3,I2,
     1  I3,I2,I3,I2,9XI2)
      IF (IT .EQ. 47) GO TO 78
      IT = 47
      GO TO 3
C
C     SET UP DATA FOR GRIDIV
   78 L  =I(3)
      ICNT = ICNT + 1
      XL  = W(1)
      XR  = W(2)
      YB  = W(3)
      YT  = W(4)
      DX  = W(5)
      DY  = W(6)
      N   = I(4)
      M   = I(5)
      IV  = I(6)
      IH  = I(7)
      NX  = I(16)
      NY  = I(17)
      IC  = I(18)
      CALL CAMRAV(IC)
      CALL GRIDIV (L,XL,XR,YB,YT,DX,DY,N,M,IV,IH,NX,NY)
```

```
DECK GRPD
C
C     CHECK TO SEE IF DATA HAS BEEN CALLED IN
      IF (NC .LT. 1) GO TO 30
C
C     TRANSFER DATA TO PLOTTING ARRAYS
      NX = I(1)
      NY = I(2)
      DO 24 J = 1,NC
      GO TO (11,12,13,14,15,16,57,58,59,60,61,62),NX
   11 X(J)     = A(J)
      GO TO 17
   12 X(J)     = B(J)
      GO TO 17
   13 X(J)     = C(J)
      GO TO 17
   14 X(J)     = D(J)
      GO TO 17
   15 X(J)     = E(J)
      GO TO 17
   16 X(J)     = F(J)
      GO TO 17
   57 X(J)     = AA(J)
      GO TO 17
   58 X(J)     = BB(J)
      GO TO 17
   59 X(J)     = CC(J)
      GO TO 17
   60 X(J)     = DD(J)
      GO TO 17
   61 X(J)     = EE(J)
      GO TO 17
   62 X(J)     = FF(J)
   17 GO TO  (18,19,20,21,22,23,63,64,65,66,67,68),NY
   18 Y(J)     = A(J)
      GO TO 24
   19 Y(J)     = B(J)
```

DECK GRPD

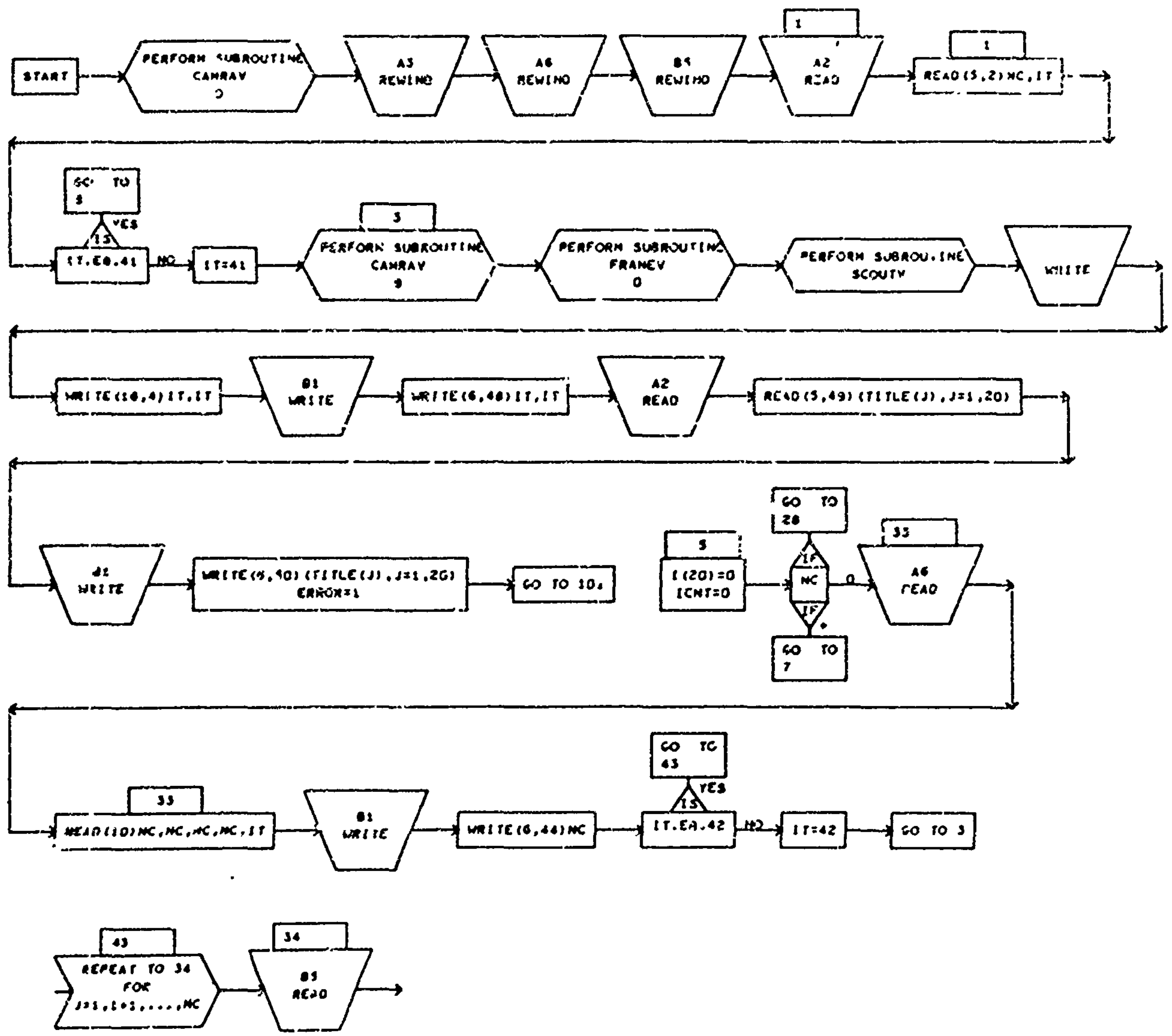
```
      GO TO 24
   20 Y(J)     = C(J)
      GO TO 24
   21 Y(J)     = D(J)
      GO TO 24
   22 Y(J)     = E(J)
      GO TO 24
   23 Y(J)     = F(J)
      GO TO 24
   63 Y(J)     = AA(J)
      GO TO 24
   64 Y(J)     = BB(J)
      GO TO 24
   65 Y(J)     = CC(J)
      GO TO 24
   66 Y(J)     = DD(J)
      GO TO 24
   67 Y(J)     = EE(J)
      GO TO 24
   68 Y(J)     = FF(J)
   24 CONTINUE
C
C     SET UP DATA FOR PLOTTING ALL POINTS THREE TIMES
C     CHECK FOR INPUT NUMBER OF POINTS TO BE PLOTTED
      IF (I(8) .GT. 0) GO TO 35
      I(8) = NC
   35 NP = I(8)
      NX = I(9)
      NY = I(10)
      IP = I(11)
      IS = I(12)
      DO 25 J = 1,3
   25 CALL APLOTV (NP,X,Y,NX,NY,IP,IS,IERR)
C
C     DUMMY CHECK TO SEE IF ANY POINTS OUTSIDE GRID
      IF (IERR .EQ. 0) GO TO 31
```

```
DECK GRPD
      IERR = 0                                                          GRPD  2160
C                                                                       GRPD  2170
C     CHECK TO SEE IF POINTS ARE TO BE CONNECTED                        GRPD  2180
   31 IF (I(13) .LT.1) GO TO 30                                         GRPD  2190
C                                                                       GRPD  2200
C     SET UP TO CONNECT DATA POINTS (3 TIMES)                           GRPD  2210
      DO 26 J = 1,3                                                     GRPD  2220
      DO 26 K = 1,NC,NX                                                 GRPD  2230
      G1        = G2                                                    GRPD  2240
      H1        = H2                                                    GRPD  2250
      IX1       = IX2                                                   GRPD  2260
      IY1       = IY2                                                   GRPD  2270
      G2        = X(K)                                                  GRPD  2280
      H2        = Y(K)                                                  GRPD  2290
      IX2       = NXV(G2)                                               GRPD  2300
      IY2       = NYV(H2)                                               GRPD  2310
      IF (K .EQ. 1) GO TO 26                                            GRPD  2320
C                                                                       GRPD  2330
C     TEST FOR OFF SCALE POINTS                                         GRPD  2340
      IF (IX1*IY1 .EQ. 0) GO TO 26                                      GRPD  2350
      IF (IX2*IY2 .EQ. 0) GO TO 26                                      GRPD  2360
      CALL LINEV (IX1,IY1,IX2,IY2)                                      GRPD  2370
   26 CONTINUE                                                          GRPD  2380
C                                                                       GRPD  2390
C     PRINT VERTICAL AND HORIZONTAL TITLES 3 TIMES                      GRPD  2400
   30 DO 36 J = 1,3                                                     GRPD  2410
   36 CALL APRNTV (0,-12,70,TITLE,8,943)                                GRPD  2420
C                                                                       GRPD  2430
C     SET UP FOR HORIZONTAL TITLE                                       GRPD  2440
      DO 27 J = 1,18                                                    GRPD  2450
   27 PRINT(J) = TITLE(J+18)                                            GRPD  2460
      DO 37 J = 1,3                                                     GRPD  2470
   37 CALL PRINTV (70,PRINT,224,8)                                      GRPD  2480
C                                                                       GRPD  2490
C     CHECK FOR HORIZONTAL LABEL CARDS                                  GRPD  2500
      IF (I(18) .LE. 0) GO TO 42                                        GRPD  2510
```

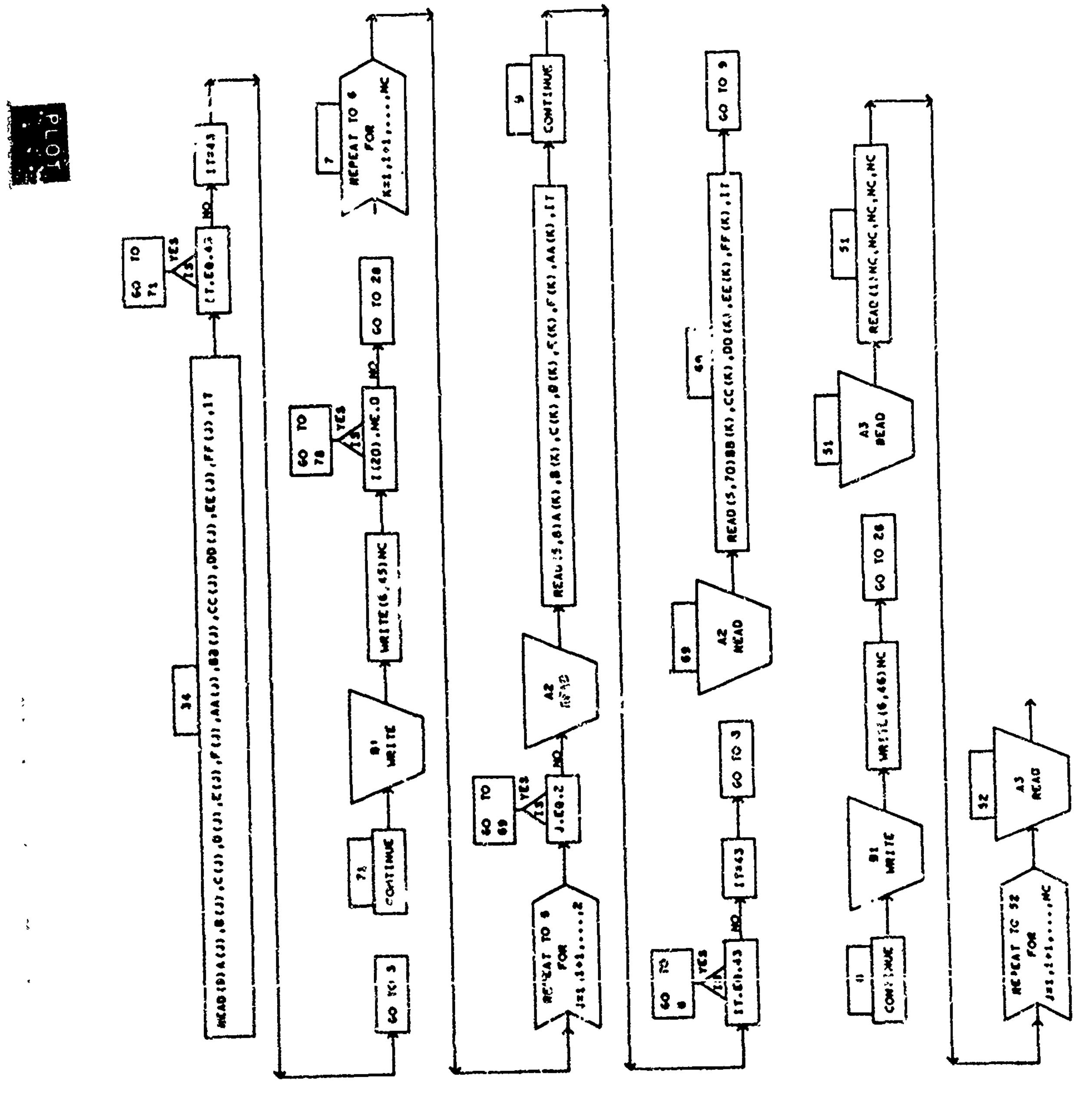
```
DECK GRPD

C                                                                   GRPD  2520
C     READ AND CHECK TYPE OF HORIZONTAL LABEL CARDS                 GRPD  2530
      J       = I(18)                                               GRPD  2540
      DO 39 K = 1,J                                                 GRPD  2550
      M       = 18*(K+2)                                            GRPD  2560
      L       = M - 17                                              GRPD  2570
      READ (5,38) (TITLE(N),N=L,M),IT                               GRPD  2580
   38 FORMAT (17A4,A2,I2)                                           GRPD  2590
       IF (IT .EQ. 48) GO TO 39                                     GRPD  2600
C                                                                   GRPD  2610
C     TYPE ERROR                                                    GRPD  2620
      IT = 48                                                       GRPD  2630
      GO TO 3                                                       GRPD  2640
   39 CONTINUE                                                      GRPD  2650
C                                                                   GRPD  2660
      PRINT ALL HORIZONTAL LABELS 3 TIMES                           GRPD  2670
      DO 41 K = 1,3                                                 GRPD  2680
      DO 41 L = 1,J                                                 GRPD  2690
      IY      = 1039 - 16*L                                         GRPD  2700
      M       = 18*(L+1)                                            GRPD  2710
      DO 40 N = 1,18                                                GRPD  2720
      II      = M + N                                               GRPD  2730
   40 PRINT(N)= TITLE(II)                                           GRPD  2740
   41 CALL PRINTV (70,PRINT,224,IY)                                 GRPD  2750
C                                                                   GRPD  2760
C     CHECK FOR WRITING ARRAYS ON TAPE 1                            GRPD  2770
   42 IF (I(19) .NE. 1) GO TO 56                                    GRPD  2780
C                                                                   GRPD  2790
C     TRANSFER ARRAYS TO TAPE 1                                     GRPD  2800
      WRITE (1) NC,NC,NC,NC,NC                                      GRPD  2810
      DO 54 J = 1,NC                                                GRPD  2820
   54 WRITE (1) A(J),B(J),C(J),D(J),E(J),F(J),AA(J),BB(J),          GRPD  2830
     1CC(J),DD(J),EE(J),FF(J)                                       GRPD  2840
      REWIND 1                                                      GRPD  2850
      WRITE (6,55) NC                                               GRPD  2860
   55 FORMAT (1H0,43X12HI JUST WROTEI4,16H CARDS ON TAPE 1 )        GRPD  2870
```

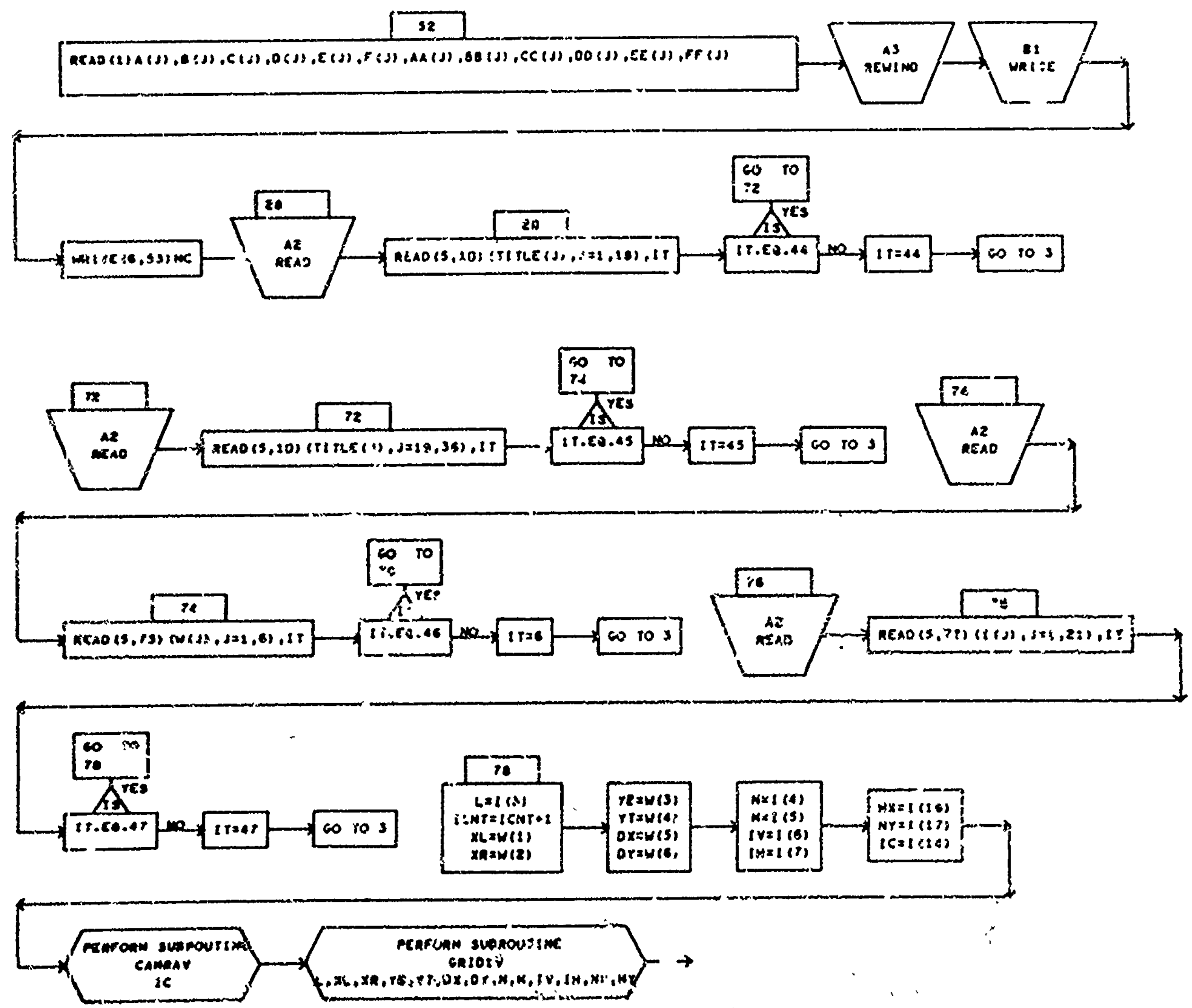
```
DECK GRPD

C
C      CHECK RETURNS OR FINISH OF PROGRAM
  56   IF (ICNT .EQ. I(20))I(15) = I(21)
       IR = I(15) + 1
       GO TO (29,28,1,33,51), IR
   29  WRITE (6,47)
   47  FORMAT (1H0,   36HJUST FINISHED PLOTTING ALL SORTS OF
     1   49HGOODIES ON TAPE 16. IF ALL GOES WELL, YOU SHOULD
     2   34HGET SOME RESULTS FROM THE SC-4020. )
       ERROR = 1
       READ (5,100) TYPE
 100   FORMAT (70X,I2)
       IF (TYPE .EQ. 99) ERROR = 0
 101   RETURN
       END
```

SUBROUTINE PLOT

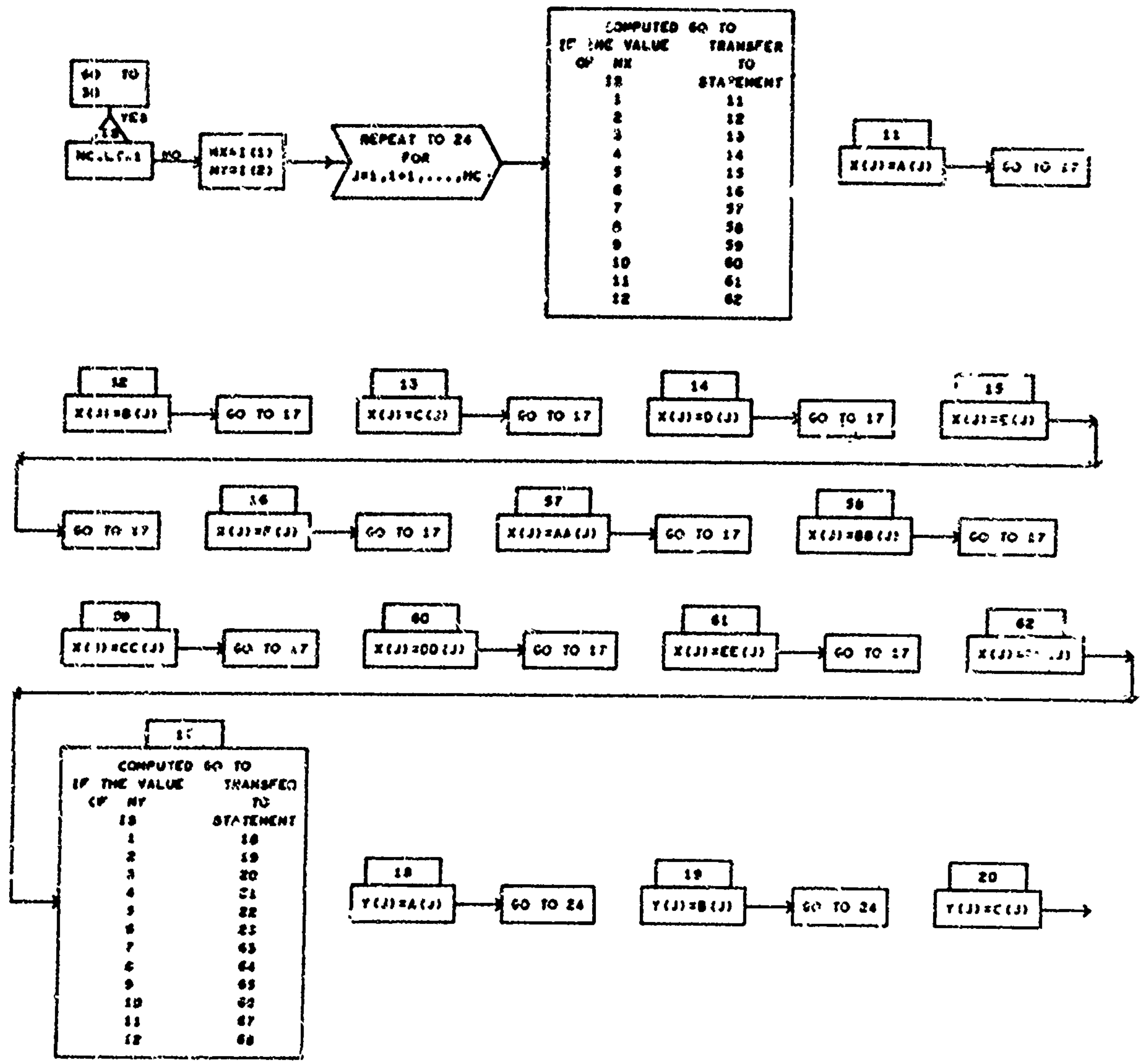


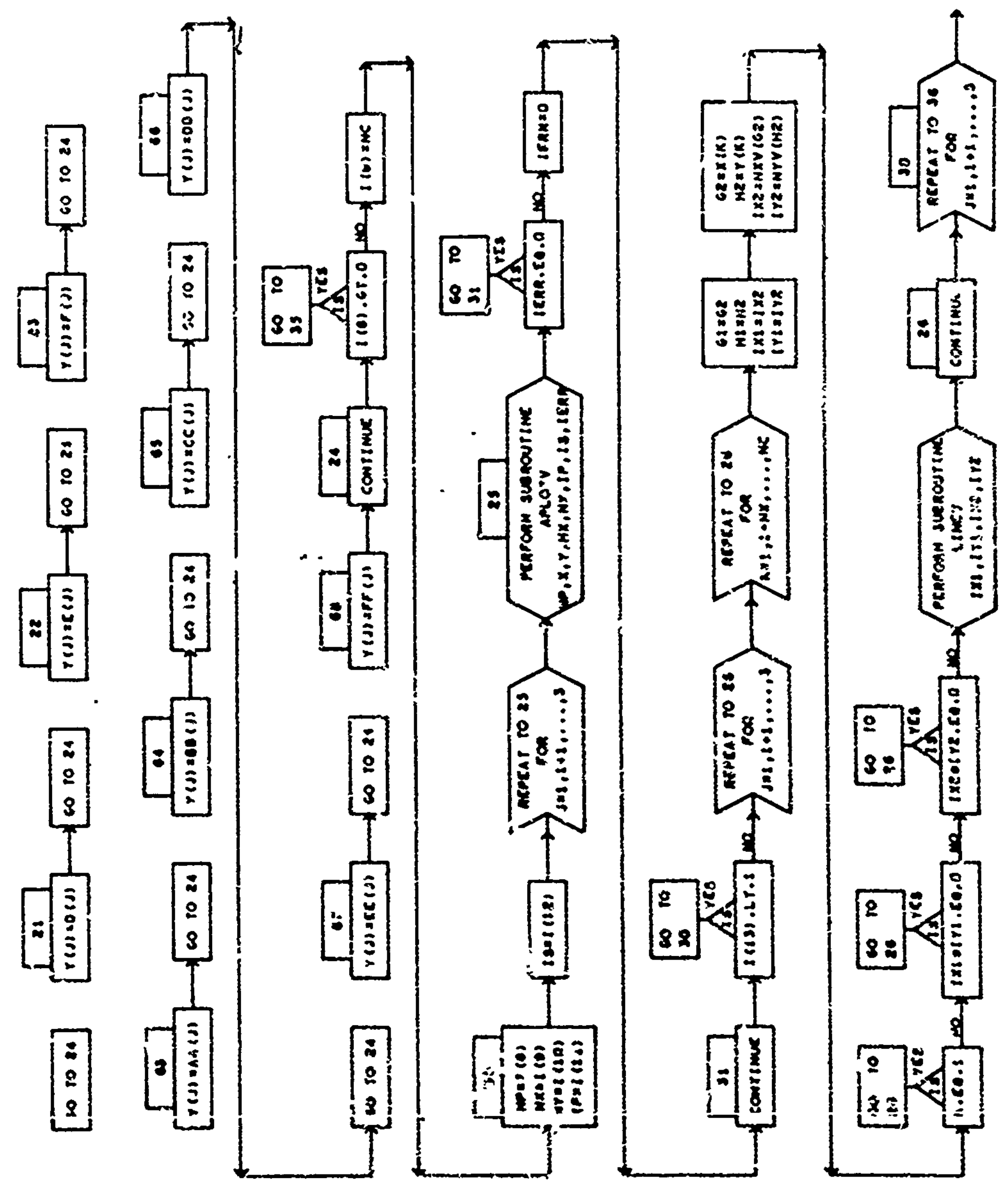
PLOT

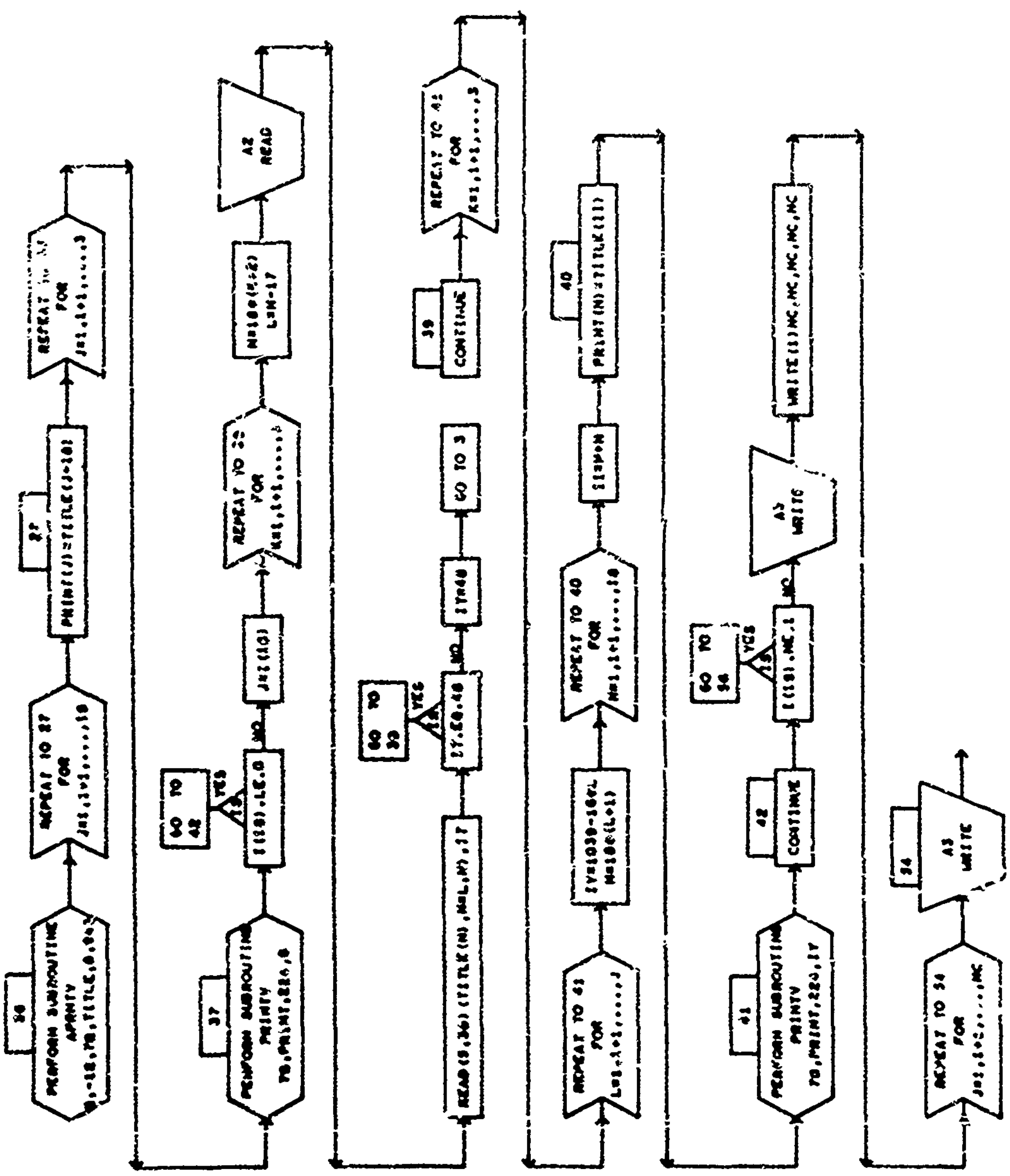

PLOT
34
READ(9)A(J),B(J),C(J),D(J),E(J),F(J),AA(J),BB(J),CC(J),DD(J),EE(J),FF(J),IT
GO TO 71
YES
IS
IT.EQ.43
NO
IT=43
GO TO 3
71
CONTINUE
B1
WRITE
WRITE(6,45)NC
GO TO 78
YES
IS
I(20).NE.0
NO
GO TO 28
7
REPEAT TO 6
FOR
K=1,1+1,....,NC
GO TO 69
YES
IS
J.EQ.2
NO
A2
READ
9
CONTINUE
GO TO 8
YES
IS
IT.EQ.43
NO
IT=43
GO TO 3
69
A2
READ
READ(5,70)BB(K),CC(K),DD(K),EE(K),FF(K),IT
GO TO 9
11
CONTINUE
B1
WRITE
GO TO 28
51
A3
READ
51
READ(1)NC,NC,NC,NC,NC
FOR
52
A3
READ

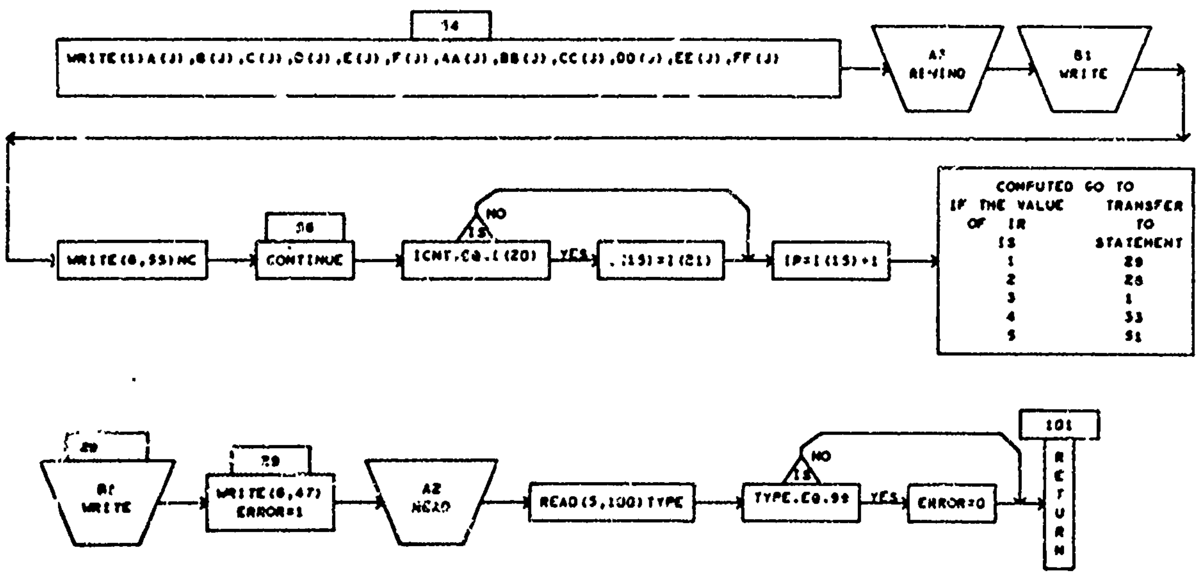


52
READ(1)A(J),B(J),C(J),D(J),E(J),F(J),AA(J),BB(J),CC(J),DD(J),EE(J),FF(J)
A3
REWIND
B1
WRITE
WRITE(6,53)NC
28
A2
READ
28
READ(5,10)(TITLE(J),J=1,18),IT
GO TO
72
YES
IS
IT.EQ.44
NO
IT=44
GO TO 3
72
A2
READ
72
READ(5,10)(TITLE(J),J=19,36),IT
GO TO
74
YES
IS
IT.EQ.45
NO
IT=45
GO TO 3
74
A2
READ
74
READ(5,73)(W(J),J=1,6),IT
GO TO
IT.EQ.46
NO
IT=6
GO TO 3
76
A2
READ
GO
78
YES
IS
IT.EQ.47
NO
IT=47
GO TO 3
78
PERFORM SUBROUTINE
CAMRAY
IC
PERFORM SUBROUTINE


GO TO 31
YES
NO
NX=I(1)
NY=I(2)
REPEAT TO 24 FOR J=1,1+1,....,NC
COMPUTED GO TO
IF THE VALUE OF NX
TRANSFER TO STATEMENT
11
X(J)=A(J)
GO TO 17
12
X(J)=B(J)
13
X(J)=C(J)
14
X(J)=D(J)
15
16
57
X(J)=AA(J)
58
X(J)=BB(J)
59
X(J)=CC(J)
60
X(J)=DD(J)
61
X(J)=EE(J)
62
IF THE VALUE OF NY
18
Y(J)=A(J)
GO TO 24
19
Y(J)=B(J)
20
Y(J)=C(J)




36
PERFORM SUBROUTINE
APRNTV
REPEAT TO 27
FOR
27
REPEAT
FOR
37
PERFORM SUBROUTINE
PRINTV
GO TO
42
YES
NO
REPEAT TO 22
FOR
A2
READ
GO TO
39
YES
NO
GO TO 3
39
CONTINUE
FOR
REPEAT TO 41
FOR
REPEAT TO 40
FOR
40
41
PERFORM SUBROUTINE
PRINTV
42
CONTINUE
GO TO
34
YES
NO
A3
WRITE
WRITE(1)NC,NC,NC,NC,NC
REPEAT TO 34
FOR
34
A3
WRITE


14
WRITE(1)A(J),B(J),C(J),D(J),E(J),F(J),AA(J),BB(J),CC(J),DD(J),EE(J),FF(J)
A? REWIND
B1 WRITE
WRITE(6,55)NC
36
CONTINUE
ICNT.EQ.I(20)
IS
NO
YES
.I15)=I(21)
IP=I(15)+1
COMPUTED GO TO
IF THE VALUE OF IR IS
TRANSFER TO STATEMENT
1 29
2 28
3 1
4 33
5 51
29
B1 WRITE
28
WRITE(6,47)
ERROR=1
A2 READ
READ(5,100)TYPE
TYPE.EQ.99
IS
NO
YES
ERROR=0
101
RETURN

PLOT

SYMBOLS USED IN SUBROUTINE PLOT

A	R	D	FIRST DATA ARRAY	PLOT
AA	R	D	SEVENTH DATA ARRAY	PLOT
B	R	D	SECOND DATA ARRAY	PLOT
BB	R	D	EIGHT DATA ARRAY	PLOT
C	R	D	THIRD DATA ARRAY	PLOT
CASE	I	C	CARD NUMBER	PLOT
CC	R	D	NINTH DATA ARRAY	PLOT
D	R	D	FOURTH DATA ARRAY	PLOT
DD	R	D	TENTH DATA ARRAY	PLOT
DX	R	U	INCREMENT BETWEEN VERTICAL GRID LINES	PLOT
DY	R	U	INCREMENT BETWEEN HORIZONAL GRID LINES	PLOT
E	R	D	FIFTH DATA ARRAY	PLOT
EE	R	D	ELEVENTH DATA ARRAY	PLOT
ERROR	I	C	ERROR FLAG	PLOT
F	R	D	SIXTH DATA ARRAY	PLOT
FF	R	D	TWELVETH DATA ARRAY	PLOT
G1	R	U	ACTUAL LOCATION ALONG X-AXIS FIRST PLOTTED POINT	PLOT
G2	R	U	ACTUAL LOCATION ALONG X-AXIS OF SECOND PLOTTED POINT	PLOT
H1	R	U	ACTUAL LOCATION ALONG Y-AXIS OF FIRST PLOTTED POINT	PLOT
H2	R	U	ACTUAL LOCATION ALONG Y-AXIS SECOND PLOTTED POINT	PLOT
I	I	D	PROGRAM CONTROL ARRAY	PLOT
IC	I	U	CAMERA SELECTION FLAG	PLOT
ICNT	I	U	PICTURE COUNTER	PLOT
IERR	I	U	NAME OF AN ERROR LOCATION.	PLOT
IH	I	U	TERM CAUSES LABEL OF EVERY IHTH HORIZONTAL GRID LINE	PLOT
II	I	U	HORIZONTAL TITLE ARRAY SUBSCRIPT	PLOT
IP	I	U	NUMBER OF CHARACTERS TO BE USED AS PLOTTING SYMBOLS	PLOT
IR	I	U	CONTROL FLAG VALUE OF WHICH DETERMINES NEXT OPERATION	PLOT
IS	I	U	PLOTTING SYMBOL CODE	PLOT
IT	I	U	TYPE OF INPUT DATA CARD	PLOT
IV	I	U	TERM CAUSES LABEL OF EVERY IVTH VERTICAL GRID LINE	PLOT
IX1	I	U	RASTER LOCATION ALONG X-AXIS OF FIRST PLOTTED POINT	PLOT
IX2	I	U	RASTER LOCATION ALONG X-AXIS OF SECOND PLOTTED POINT	PLOT
IY	I	U	VERTICAL RASTER LOCATION OF HORIZONTAL TITLES	PLOT
IY1	I	U	RASTER LOCATION ALONG Y-AXIS SECOND PLOTTED POINT	PLOT
IY2	I	U	RASTER LOCATION ALUNG Y-AXIS OF SECOND PLOTTED POINT	PLOT

SYMBOLS USED IN SUBROUTINE PLOT

J	I	U	MULTI-PURPOSE INDEX	PLOT
K	I	U	MULTI-PURPOSE INDEX	PLOT
L	I	U	MULTI-PURPOSE INDEX AND FILM ADVANCE FLAG	PLOT
M	I	U	MULTI-PURPOSE INDEX AND EMPHASIS FLAG FOR HORIZ. GRID LINES	PLOT
N	I	U	MULTI-PURPOSE INDEX AND EMPHASIS FLAG FOR VERT. GRID LINES	PLOT
NC	I	U	NUMBER OF DATA POINTS	PLOT
NP	I	U	NUMBER OF POINTS TO BE PLOTTED	PLOT
NX	I	U	SUBSCRIPT INCREMENT OF X-ARRAY DATA TO BE PLOTTED	PLOT
NY	I	U	SUBSCRIPT INCREMENT OF X-ARRAY DATA TO BE PLOTTED.	PLOT
PAGE	I	C	PAGE NUMBER	PLOT
PRINT	R	D	PRINTING ARRAY	PLOT
TITLE	R	D	ABSISSA AND ORDINATE TITLES, AND HORIZONTAL TITLE ARRAY	PLOT
TITLE2	R	C	DUMMY TITLE ARRAY	PLOT
TYPE	I	U	CARD TYPE	PLOT
W	R	D	GRID INFORMATION ARRAY	PLOT
X	R	D	PLOTTING ARRAY, LOCATION ALONG X-AXIS	PLOT
XL	R	U	LEFT-MOST LIMIT OF THE GRID ON X-AXIS	PLOT
XR	R	U	RIGHT-MOST LIMIT OF THE GRID ON X-AXIS	PLOT
Y	R	D	PLOTTING ARRAY, LOCATION ALONG Y-AXIS	PLOT
YB	R	U	BOTTOM MOST LIMIT OF THE GRID ON Y-AXIS	PLOT
YT	R	U	TOP MOST LIMIT OF THE GRID ON Y-AXIS	PLOT

SYMBOLS USED IN SUBROUTINE PLOT

J	I	U	MULTI-PURPOSE INDEX	PLOT
K	I	U	MULTI-PURPOSE INDEX	PLOT
L	I	U	MULTI-PURPOSE INDEX AND FILM ADVANCE FLAG	PLOT
M	I	U	MULTI-PURPOSE INDEX AND EMPHASIS FLAG FOR HORIZ. GRID LINES	PLOT
N	I	U	MULTI-PURPOSE INDEX AND EMPHASIS FLAG FOR VERT. GRID LINES	PLOT
NC	I	U	NUMBER OF DATA POINTS	PLOT
NP	I	U	NUMBER OF POINTS TO BE PLOTTED	PLOT
NX	I	U	SUBSCRIPT INCREMENT OF X-ARRAY DATA TO BE PLOTTED	PLOT
NY	I	U	SUBSCRIPT INCREMENT OF X-ARRAY DATA TO BE PLOTTED.	PLOT
PAGE	I	C	PAGE NUMBER	PLOT
PRINT	R	D	PRINTING ARRAY	PLOT
TITLE	R	D	ABSISSA AND ORDINATE TITLES, AND HORIZONTAL TITLE ARRAY	PLOT
TITLE2	R	C	DUMMY TITLE ARRAY	PLOT
TYPE	I	U	CARD TYPE	PLOT
W	R	D	GRID INFORMATION ARRAY	PLOT
X	R	D	PLOTTING ARRAY, LOCATION ALONG X-AXIS	PLOT
XL	R	U	LEFT-MOST LIMIT OF THE GRID ON X-AXIS	PLOT
XR	R	U	RIGHT-MOST LIMIT OF THE GRID ON X-AXIS	PLOT
Y	R	D	PLOTTING ARRAY, LOCATION ALONG Y-AXIS	PLOT
YB	R	U	BOTTOM MOST LIMIT OF THE GRID ON Y-AXIS	PLOT
YT	R	U	TOP MOST LIMIT OF THE GRID ON Y-AXIS	PLOT

32. SUBROUTINE SLABD (DECK SLBA)

This routine generates the element data for a simple slab delta vehicle.

a. Algorithm

Read in Slab Delta Title Card (Type 50), and the Configuration Control Card (Type 51). If required read in the Thickness Correction Cards (Type 52 and 53). Read in a Cross-Section Data Card (Type 54) and calculate the element data for this X-station. Write the card images on the regular output Tape 6 if required and also on the geometry storage Tape 8. Continue until all the X-station cards have been read.

b. Input/Output

Slab Delta Title Card (Type 50), Slab Delta Sweep Card (Type 51), and the Slab Delta Station Data Card (Type 54). If ITOC is equal to 1 the Thickness Correction Cards (Type 52 and 53) are also input.
The card images of the element data are written on the normal output Tape 6 if IPRINT equals 1, and also on the geometry storage Tape 8.

c. Error

An error condition occurs when a card type number is wrong.

d. Subroutines Required

TTABLE

e. Argument List

None

f. Length

7584 bytes

DECK SLBA

```
      SUBROUTINE SLABD
C
C*****************************************************************************
C **** THIS PROGRAM PREPARES SURFACE ELEMENT DATA OF ANALYTICAL SHAPES
C***** FOR USE IN THE FORCE PROGRAM.  THIS PROGRAM GENERATES THE
C***** SURFACE DATA CARDS AND STORES THEM ON TAPE 8.
C*****************************************************************************
C
      DIMENSION  TITLE(15),ZT(300),ZB(300),CARD(20)
C
      COMMON  CASE,TITLE,PAGE,ERROR
      INTEGER  STAT,STATT,CASE,TYPE,SEQ,PAGE,ROWS,THETAB,THETAF,STATA,
     1  LAST,LAST2,LAST3,PAGE,ERROR
C
C  SET COUNTERS
    1 LINE = 100
      SEQ = 1
      IP = 1
      STATA= 2
      TYPE = 3

C  READ IN TITLE CARD
      READ (5,100) (TITLE(L),L=1,15),LAST,CASE,ITYPE
  100 FORMAT(14A4,1X3,I1,5XI3,2XI2)
      IF (ITYPE .NE. 50) GO TO 300
C
C
C  THE FOLLOWING STATEMENTS TO STATEMENT 5 MAY BE ALTERED OR
C  REPLACED FOR OTHER ANALYTICAL SHAPES
C
C  BLUNT SLAB DELTA WING SURFACE DATA GENERATION
C  READ INPUT CASE DATA CARD
      READ (5,106) SWEEP,RNOSE,THETAB,THETAT,NOSPAN,ITOC,MODE,
     1  IREW8,IOBSP,IPRINT,ITYPE
  106 FORMAT (2F10.0,3I3,2I1,4XI1,5XI1,5X,I1,22XI2)
```

A-438

```
DECK SLBA

      IF (ITYPE .NE. 51) GO TO 300
      IF (IREW8 .EQ. 0) REWIND 8
      WRITE (6,60) SWEEP,RNOSE
   60 FORMAT (1H1,15X,42HSLAB DELTA GEOMETRY DATA WILL BE GENERATED ,/
     1 1H ,20X7HSWEEP =F6.2,5X,7HRNOSE =F9.3)
C
C  CHECK IF SPANWISE Z FACTOR DATA IS INPUT
      IF (ITOC .EQ. 0) GO TO 21
C
C  READ IN Z FACTOR DATA
      I = -4
   30 I = I + 5
      READ (5,107) ZT(I),ZT(I+1),ZT(I+2),ZT(I+3),ZT(I+4),LAST2,ITYPE
  107 FORMAT (5F10.0,9XI1.10XI2)
      IF (ITYPE .NE. 52) GO TO 300
      IF (LAST2 .EQ. 0) GO TO 30
      I = -4
   31 I = I + 5
      READ (5,107) ZB(I),ZB(I+1),ZB(I+2),ZB(I+3),ZB(I+4),LAST2,ITYPE
      IF (ITYPE .NE. 53) GO TO 300
      IF (LAST2 .EQ. 0) GO TO 31
C  SET UP INITIAL DATA
   21 DELTHB = (90.0 / FLOAT(THETAB)) / 57.2957795
C
      DELTHT = (90.0 / FLOAT(THETAT)) / 57.2957795
      THETAO = - DELTHB * FLOAT(NOSPAN)
      THEMAX = 180.0/57.2957795 + DELTHT * FLOAT(NOSPAN)
      N = 2
      I = 1
      SWEEP = SWEEP / 57.2957795
C
C  READ IN SECTION DATA CARD
   13 READ (5,108) XB,DELZ,TOPTC,BOTTC,ITCP,LAST3,ITYPE
  108 FORMAT (4F10.0,1XI1,17XI1,10XI2)
      IF (ITYPE .NE. 54) GO TO 300
C
```

DECK SLBA

```
      IF (LAST3 .EQ. 1) I = N                                           SLBA  0720
      XB = - XB                                                         SLBA  0730
C                                                                       SLBA  0740
      YLECL =  (XB - RNOSE)*COS(SWEEP)/SIN(SWEEP)                       SLBA  0750
C                                                                       SLBA  0760
C  START OF ANGULAR LOOP                                                SLBA  0770
      J = 0                                                             SLBA  0780
      ISIDE = 0                                                         SLBA  0790
   14 J = J + 1                                                         SLBA  0800
C                                                                       SLBA  0810
C   CHECK IF THIS POINT IS A TOP OR A BOTTOM POINT AND SET FLAG         SLBA  0820
   42 IF (ISIDE .EQ. 1) GO TO 53                                        SLBA  0830
      THETA = THETAO + DELTHB * (FLOAT(J)-1.0)                          SLBA  0840
      IF (THETA .GT. 1.5708    ) GO TO 44                               SLBA  0850
      GO TO 43                                                          SLBA  0860
C                                                                       SLBA  0870
   44 ISIDE = 1                                                         SLBA  0880
      THETA = THETAO + DELTHB * (FLOAT(J)-2.0)                          SLBA  0890
      THESID = (90.0/57.2957795 + DELTHT) - DELTHT * FLOAT(J)           SLBA  0900
      GO TO 43                                                          SLBA  0910
   53 THETA = THESID + DELTHT * (FLOAT(J)-1.0)                          SLBA  0920
C                                                                       SLBA  0930
C  CHECK IF SECTION HAS BEEN COMPLETED                                  SLBA  0940
   43 IF (THETA .GT. (THEMAX+0.01)) GO TO 2                             SLBA  0950
C                                                                       SLBA  0960
      IF (THETA.GT.-0.0001 .AND. THETA.LT. 3.1416    ) GO TO 16         SLBA  0970
C                                                                       SLBA  0980
      IF (YLECL .GT. 0.0) GO TO 15                                      SLBA  0990
C  SPHERE FLAT SECTIONS                                                 SLBA  1000
      YA = 0.0                                                          SLBA  1010
      RADIUS = SQRT(2.0*RNOSE*XB - XB*XB)                               SLBA  1020
      ZA = RADIUS                                                       SLBA  1030
      IF (THETA .LT. 0.0) ZA = -RADIUS                                  SLBA  1040
      GO TO 19                                                          SLBA  1050
C  FLAT SECTIONS                                                        SLBA  1060
   15 ZA = RNOSE                                                        SLBA  1070
```

```
DECK SLBA

      IF (THETA .LT. 0.0) ZA = -RNOSE
      IF (THETA .GT. 1.5708    ) GO TO 17
C  BOTTOM FLAT
      YA = (FLOAT(J-1) / (FLOAT(NOSPAN)))* YLECL
      GO TO 19
C  TOP FLAT
   17 YA = ((THEMAX - THETA) / (THEMAX - 3.14159265) ) * YLECL
      IF (ITOP .EQ. 1) YA = YLECL
      GO TO 19
C
C  SPHERE OR LEADING EDGE
   16 IF (MODE.EQ.3 .AND. ISIDE.EQ.1) GO TO 19
      IF (XB .GE. RNOSE) GO TO 18
      C = (RNOSE-XB) *SIN(SWEEP) / COS(SWEEP)
      RADIUS = SQRT(2.0*RNOSE*XB - XB*XB)
      YA = RADIUS * SIN(THETA)
      IF (YA .GT. C) GO TO 18
      ZA = -RADIUS * COS(THETA)
      GO TO 19
C
   18 YA = YLECL + (RNOSE*SIN(THETA) / SIN(SWEEP))
      ZA = -RNUSE * COS(THETA)
C
C
   19 A =RNOSE * (1.0 - COS(SWEEP))
      IF (XB .LT.A) YLE = SQRT(2.0*RNOSE*XB - XB*XB)
      IF (XB .GE.A) YLE = YLECL +(RNOSE/SIN(SWEEP))
      IF (ISIDE .EQ. 0) ZA = ZA*BOTTC
      IF (MODE .GT. 1) GO TO 45
      IF (ISIDE .EQ. 1) ZA = ZA*TOPTC
       GO TO 46
C  CHECK TOP OR BOTTOM FOR MODES 2 AND 3
   45 IF (ISIDE .EQ. 1) GO TO 47
      ZA = ZA + DELZ
      GO TO 46
   47 IF (MODE .EQ. 3) GO TO 48
```

DECK SLBA

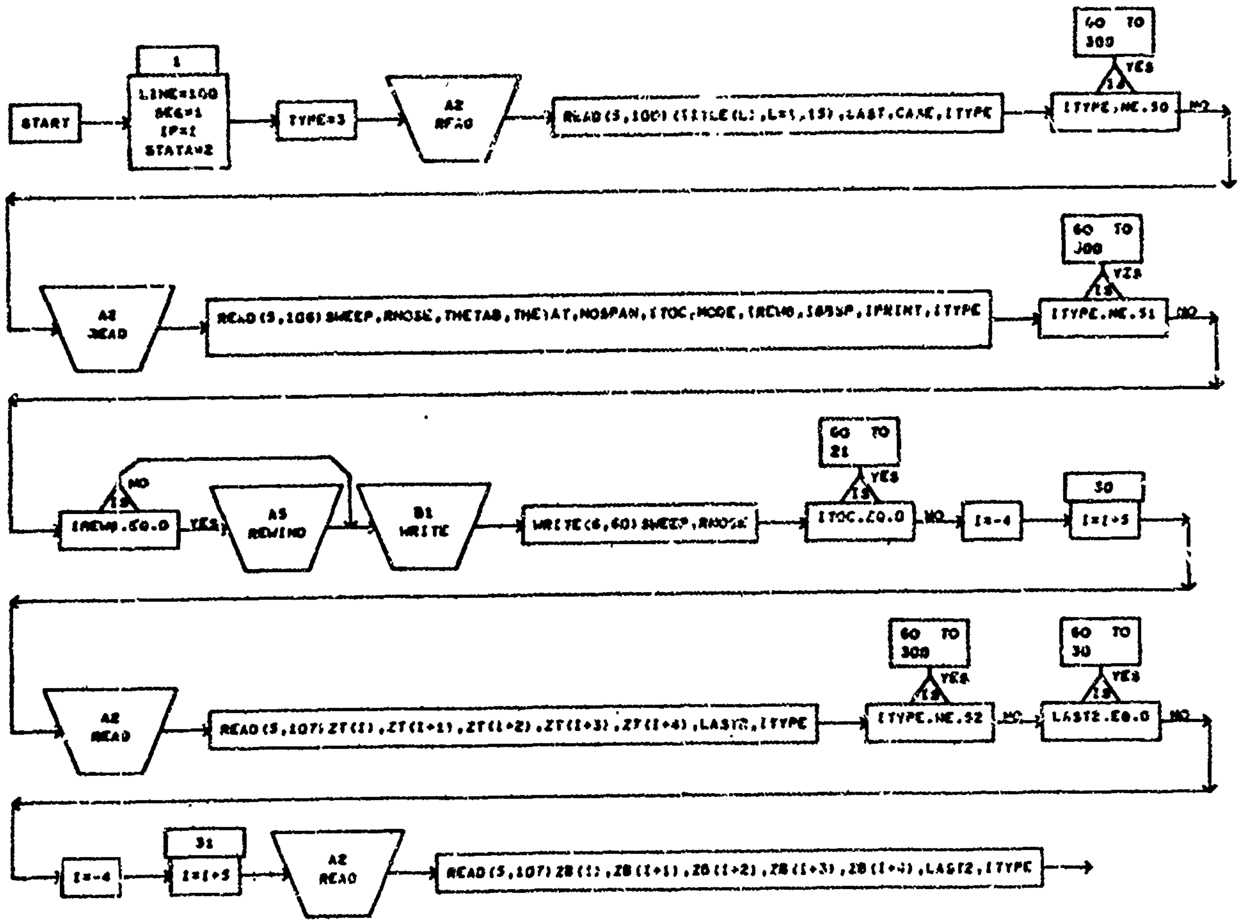
```
      ZA = ZA * TOPTC
      GO TO 46
   48 THETA2 = THETA - 1.57079633
      IF (THETA .GT. 3.1416    ) THETA2 = 1.57079633
       AA = YLE
      IF (XB .LE. RNOSE) BB = TOPTC*SQRT(2.0*RNOSE*XB - XB*XB) - DELZ
      IF (XB .GT. RNOSE) BB = RNOSE * TOPTC - DELZ
       R = AA*BB / SQRT(BB*BB*COS(THETA2)*COS(THETA2)
     1      + AA*AA*SIN(THETA2)*SIN(THETA2))
      YA = R * COS(THETA2)
      ZA = R * SIN(THETA2) + DELZ
C
C  CHECK IF SPAN CORRECTION IS TO BE MADE
   46 IF (ITOC .EQ. 0) GO TO 50
C  CORRECT THICKNESS
      PSPAN = YA / YLE
      DUMMY = 0.0
      IF (THETA .GT. 1.5708    ) GO TO 40
      CALL TTABLE (PSPAN,ZFACT,DUMMY,XB,ZB,BDOT,BDOT2)
      GO TO 41
   40 CALL TTABLE (PSPAN,ZFACT,DUMMY,ZB,ZT,BDOT,BDCT2)
   41 ZA = ZA * ZFACT
C
C
C  CORRECT FOR LEADING EDGE CENTER LINE SHIFT AND CHANGE SIGN ON XA
   50 IF (MODE .EQ. 1) ZA = ZA + DELZ
      XA = -XB
      THCHK = THEMAX - 0.01
      IF (I.EQ.N .AND. THETA.GT.THCHK) M = J
C
C  CHECK IF LAST POINT OF CASE HAS BEEN REACHED (I=N AND J=M)
      IF (I.EQ.N .AND. J.EQ.M)  STATA = 3
C
C  CHECK ON LEFT OR RIGHT DATA POINT POSITION
    5 GO TO (7,9), IP
C
```

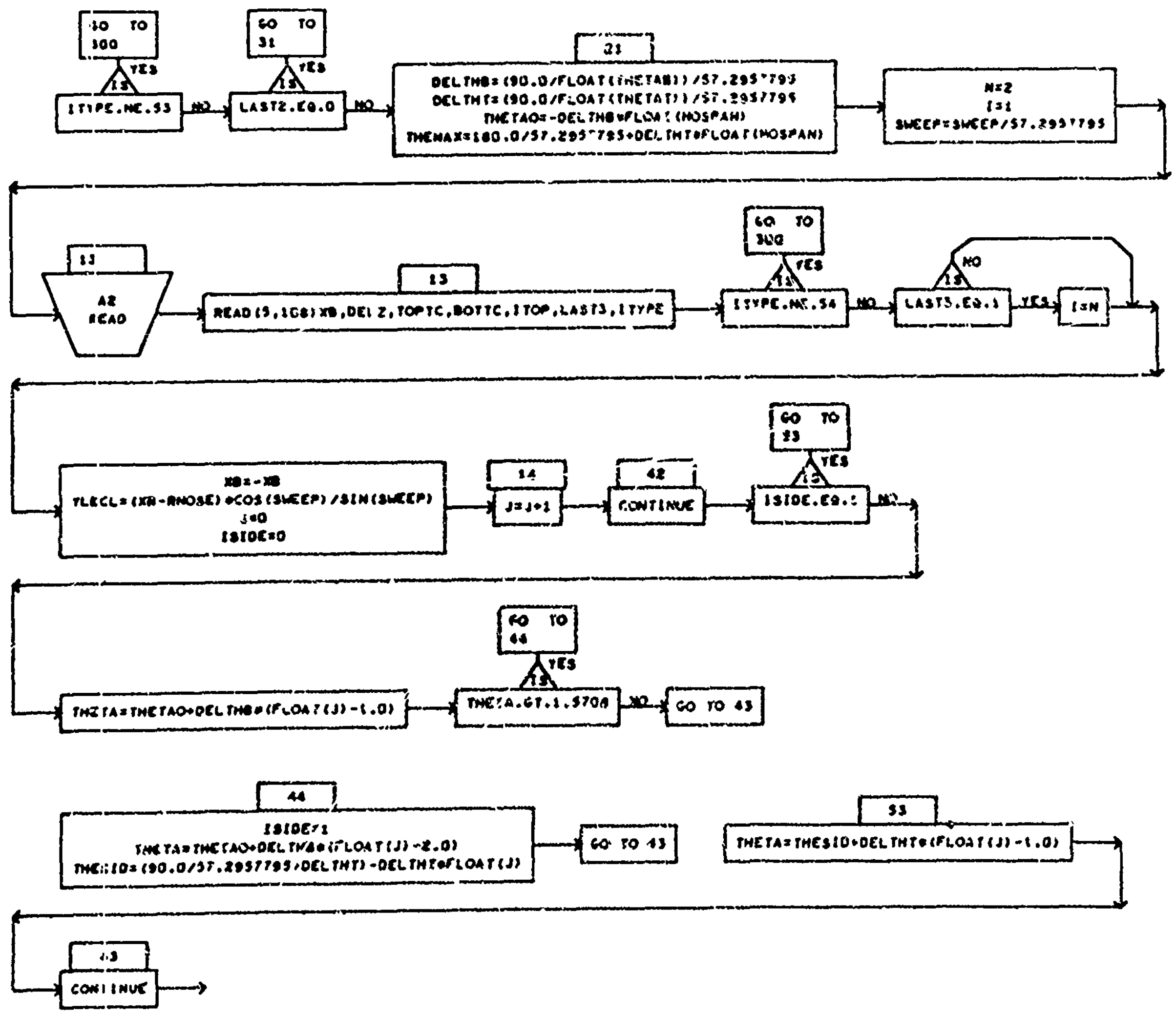
```
DECK SLBA

C  SET UP DATA FOR LEFT SIDE PRINTING AND PUNCHING
   7  X     = XA
      Y     = YA
      Z     = ZA
      STAT  = STATA
C
C  CHANGE PRINT POSITION FLAG TO RIGHT SIDE PRINT
      IP = 2
      IF (STATA.EQ.3) GO TO 12
      GO TO 3
C
C  SET UP DATA FOR RIGHT SIDE PRINTING AND PUNCHING
   8  XX    = XA
      YY    = YA
      ZZ    = ZA
      STATT= STATA
C
C  CHANGE PRINT POSITION FLAG TO LEFT SIDE PRINT
      IP = 1
C
C  CHECK LINE COUNT AND HEADER REQUIREMENT
  12  IF (IPRINT .EQ. 0) GO TO 11
      IF (LINE.LT.50) GO TO 11
C
C  PRINT HEADER AT TOP OF PAGE
      WRITE (6,101) CASE, (TITLE(L) ,L=1,15), PAGE
 101  FORMAT (1H1,5X,24HSLAB DELTA GEOMETRY DATA,/
     1  1H0,5X,6H  CASE,I5,19X,14A4,1A3,5X,5HPAGE I4,/1H0,5X
     2  1HX9X1HY9X1HZ4X1HS5X1HX9X1HY8X1HZ5X1HS18H CASE  TYPE    SEQ  )
C
C  STEP PAGE NUMBER
      PAGE = PAGE + 1
      LINE = 5
C
C  CHECK IF THIS IS A PARTIAL CARD CONDITION
  11  IF (STAT.EQ.3.AND.IP.EQ.2) GO TO 9
```

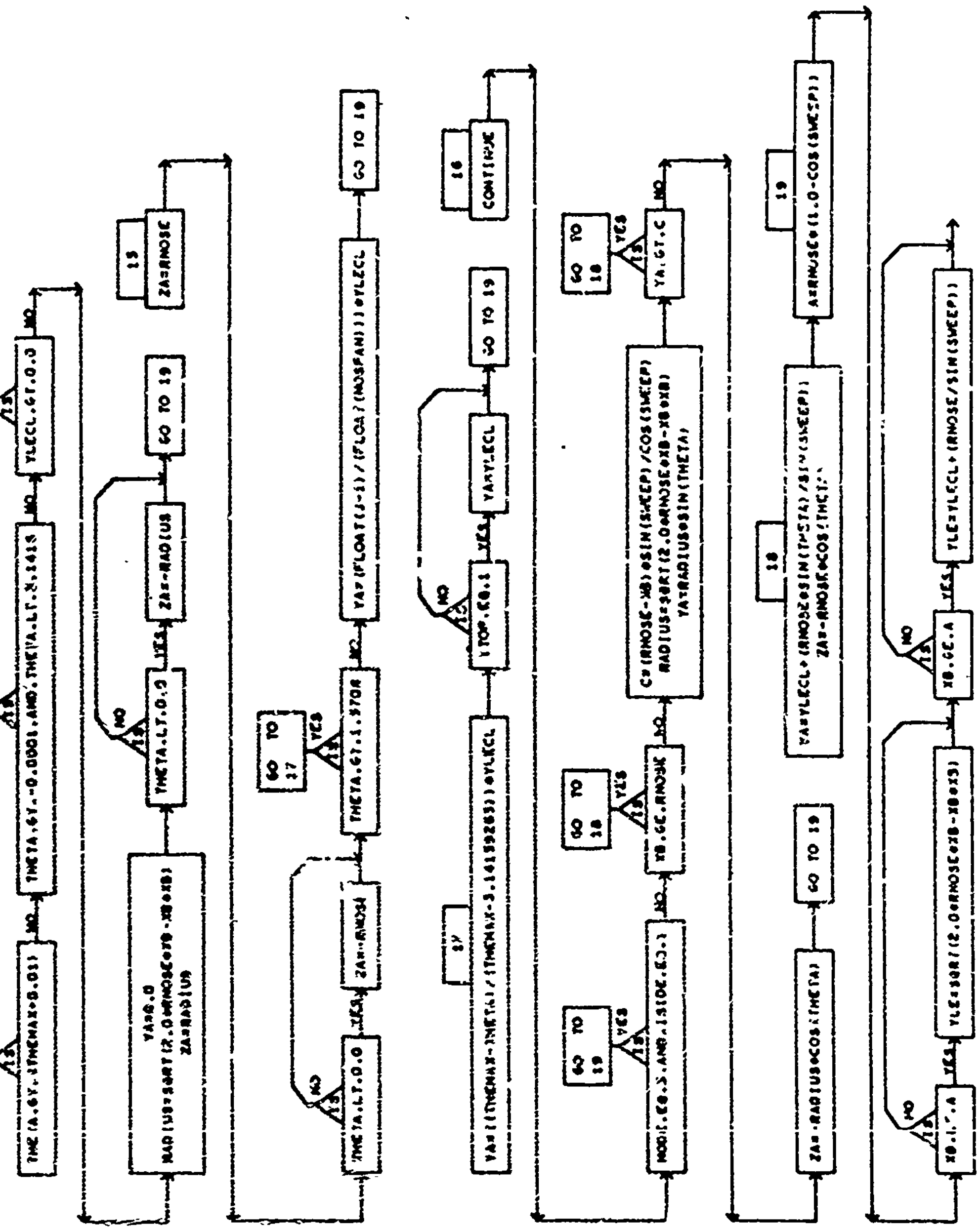
```
DECK SLBA

      IF (IPRINT .EQ. 0) GO TO 61
C
C  PRINT OUTPUT DATA FOR ONE CARD (BOTH LEFT AND RIGHT SIDE)
      WRITE (6,102) X,Y,Z,STAT,XX,YY,ZZ,STATT,CASE,TYPE,SEQ
  102 FORMAT (1H0,3F10.4,I1,3F10.4,I1,I6,3X,I1,4HAERO,I4)
      LINE = LINE + 2
C
C  WRITE DATA ON PUNCH TAPE 8 (FULL CARD)
   61 WRITE (8,103) X,Y,Z,STAT,XX,YY,ZZ,STATT,CASE,TYPE,SEQ
  103 FORMAT (3F10.4,I1,3F10.4,I1,I6,3X,I1,4HAERO,I4)
      GO TO 10
C
C  PRINT OUTPUT FOR ONE CARD (LEFT SIDE ONLY)
    9 IF (IPRINT .EQ. 0) GO TO 62
      WRITE (6,104) X,Y,Z,STAT,CASE,TYPE,SEQ
  104 FORMAT (1H0,3F10.4,I1,31X,I6,3X,I1,4HAERO,I4)
C
C  WRITE DATA ON PUNCH TAPE 8  (LEFT PART OF CARD ONLY)
   62 WRITE (8,105) X,Y,Z,STAT,CASE,TYPE,SEQ
  105 FORMAT (3F10.4,I1,31X,I6,3X,I1,4HAERO,I4)
C
C  STEP SEQUENCE COUNTER
   10 SEQ = SEQ + 1
C
C
C  END OF THETA DO LOOP - CHANGE STATUS TO 0 FOR NEXT POINT
    3 IF (STATA .EQ. 3) GO TO 52
      STATA = 0
      GO TO 14
C
C
C  END OF CROSS SECTION ROW DO LOOP - CHANGE STATUS TO 1 FOR NEXT ROW
    2 STATA = 1
      GO TO 13
C
C
```

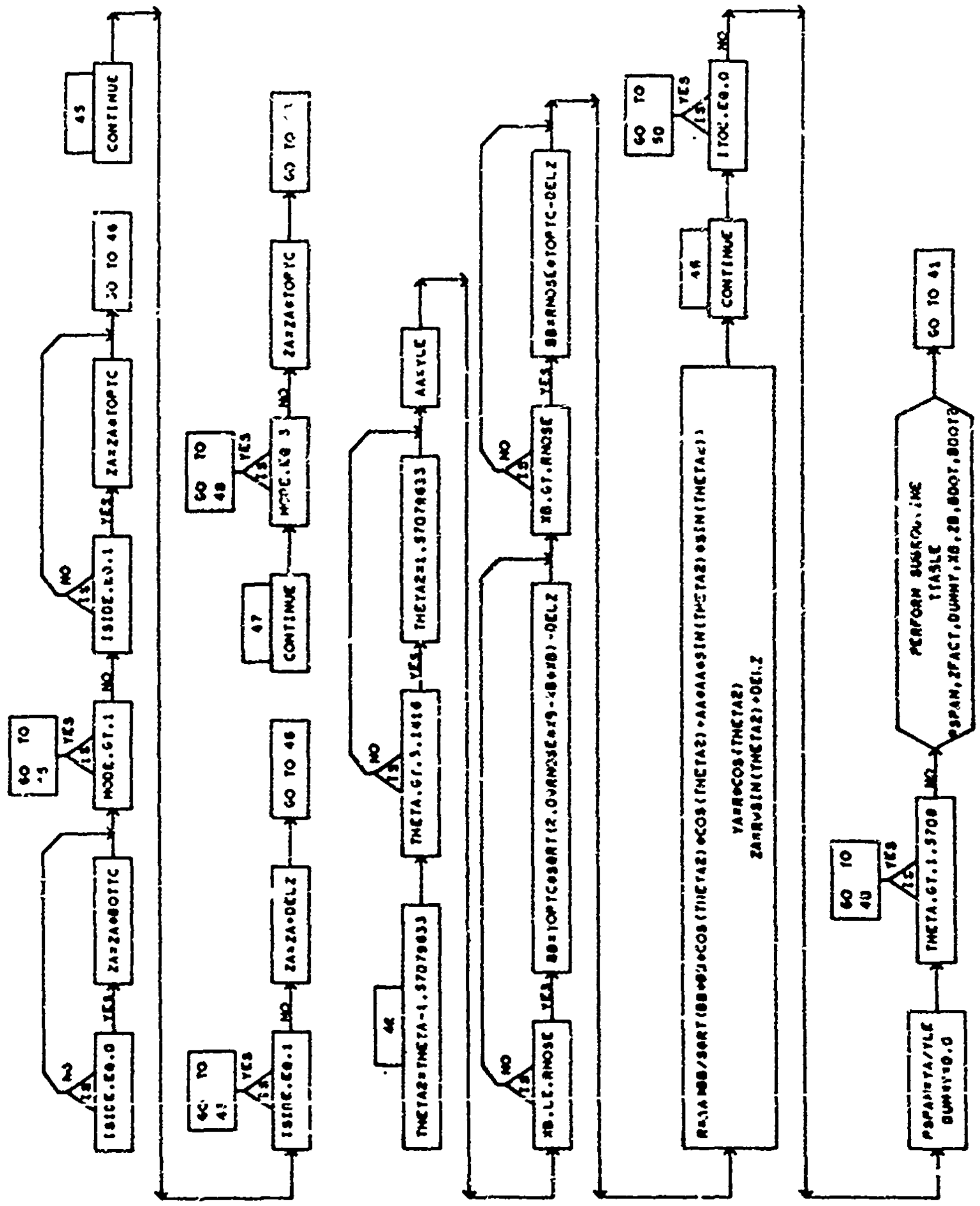
```
DECK SLBA

C  CHECK IF LAST CASE HAS BEEN REACHED                                    SLBA  2520
  52  IF (LAST .NE. 1) GO TO 1                                            SLBA  2530
C                                                                         SLBA  2540
C  LAST CASE HAS BEEN COMPLETED SO WRITE END OF FILE ON PUNCH TAPE 8     SLBA  2550
      WRITE (8,500)                                                       SLBA  2560
 500  FORMAT (12H**BLANK CARD,68X)                                        SLBA  2570
      END FILE 8                                                          SLBA  2580
      BACKSPACE 8                                                         SLBA  2590
      BACKSPACE 8                                                         SLBA  2600
      IF (I8BSP .EQ. 0) GO TO 501                                         SLBA  2610
      SEQ = SEQ - 1                                                       SLBA  2620
      DO 502 III=1,SEQ                                                    SLBA  2630
 502  BACKSPACE 8                                                         SLBA  2640
 501  ERROR = 1                                                           SLBA  2650
      READ (5,200) TYPE                                                   SLBA  2660
 200  FORMAT (70X,I2)                                                     SLBA  2670
      IF (TYPE .EQ. 99) ERROR = 0                                         SLBA  2680
      RETURN                                                              SLBA  2690
 300  ERROR = 1                                                           SLBA  2700
      WRITE (6,301)                                                       SLBA  2710
 301  FORMAT (1H0,47H*****YOU HAVE MADE AN ERROR EITHER IN CARD TYPE      SLBA  2720
     1 49H INDICATION OR CARD ORDER - CHECK YOUR CARDS*****  )            SLBA  2730
      READ (5,302) (CARD(II),II=1,20)                                     SLBA  2740
 302  FORMAT (20A4)                                                       SLBA  2750
      WRITE (6,803) (CARD(II),II=1,20)                                    SLBA  2760
 803  FORMAT (1H0,45H THE CARD LOCATED JUST BEFORE THE CARD LISTED        SLBA  2770
     1 18H BELOW IS IN ERROR,/1H ,10X,20A4)                               SLBA  2780
      RETURN                                                              SLBA  2790
      END                                                                 SLBA  2800
```

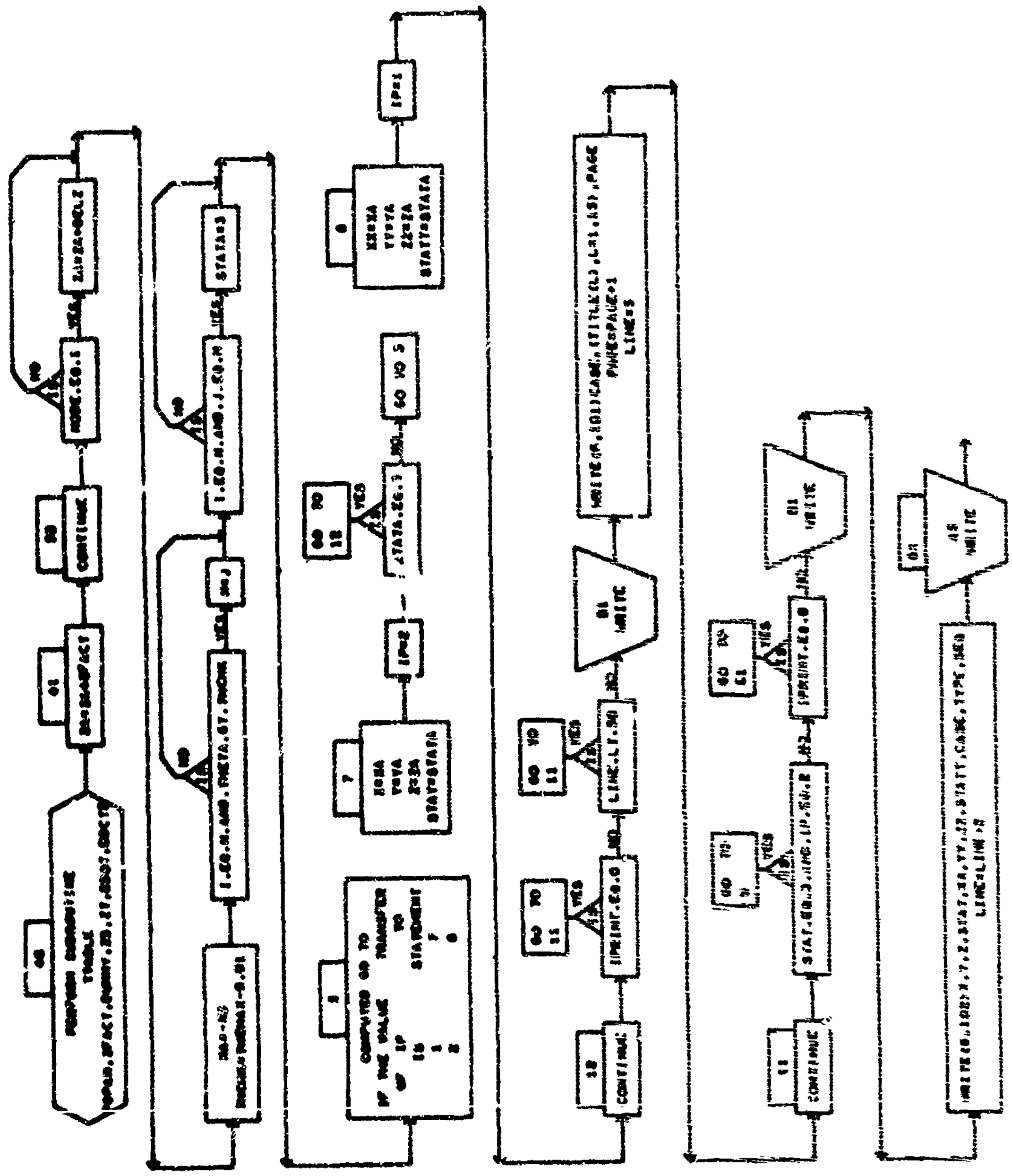
SUBROUTINE SLAG3

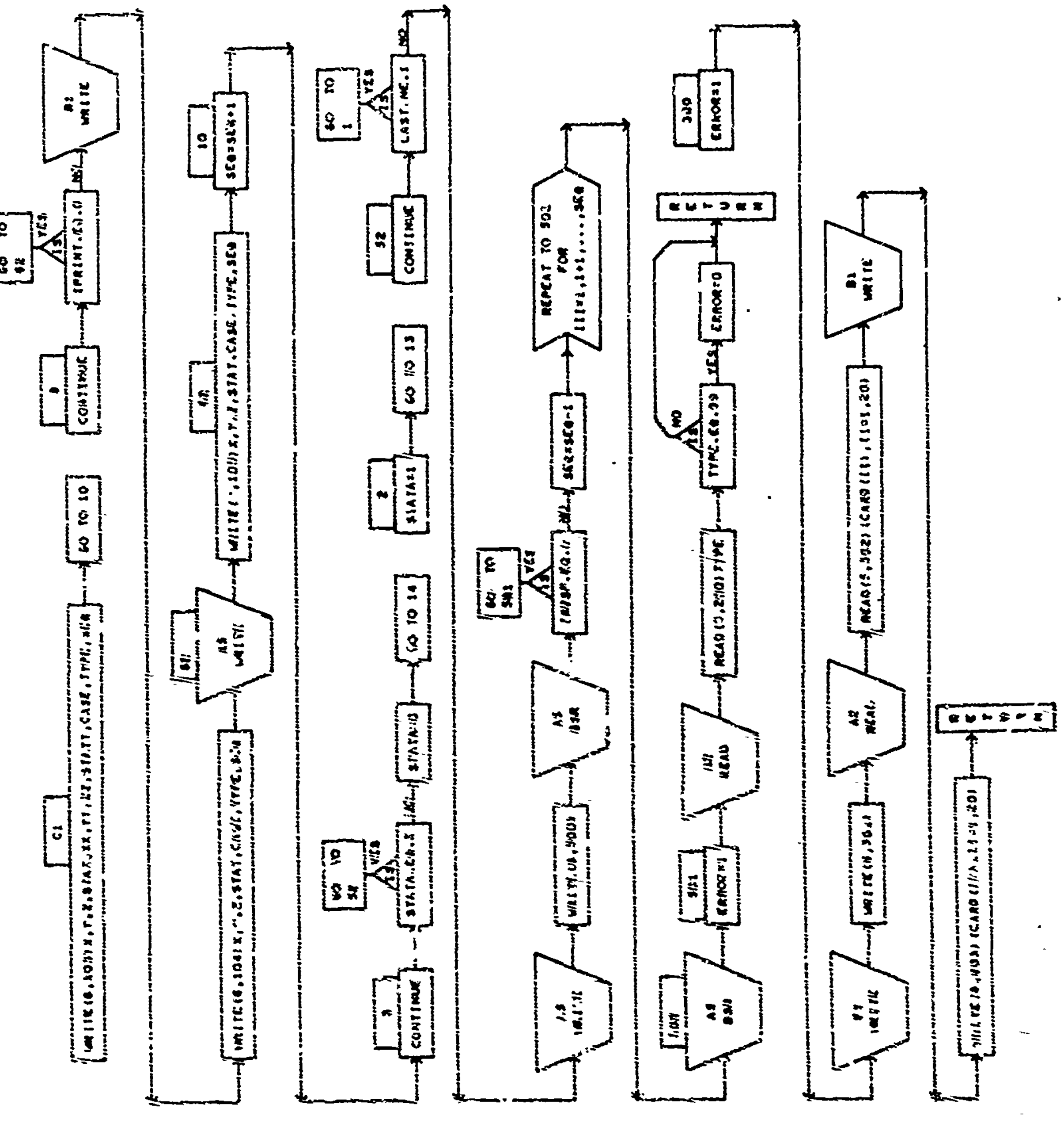



GO TO 300
YES
13
ITYPE.NE.53
NO
GO TO 31
YES
13
LAST2.EQ.0
NO
21
DELTHB=(90.0/FLOAT(THETAB))/57.2957795
DELTHT=(90.0/FLOAT(THETAT))/57.2957795
THETAO=-DELTHB*FLOAT(NOSPAN)
THEMAX=180.0/57.2957795+DELTHT*FLOAT(NOSPAN)
N=2
I=1
SWEEP=SWEEP/57.2957795
11
A2
READ
13
READ(5,108)XB,DELZ,TOPTC,BOTTC,ITOP,LAST3,ITYPE
GO TO 300
YES
13
ITYPE.NE.54
NO
NO
13
LAST3.EQ.1
YES
I=N
XB=-XB
YLECL=(XB-RNOSE)*COS(SWEEP)/SIN(SWEEP)
J=0
ISIDE=0
14
J=J+1
42
CONTINUE
GO TO 53
YES
13
ISIDE.EQ.1
NO
THETA=THETAO+DELTHB*(FLOAT(J)-1.0)
GO TO 44
YES
13
THETA.GT.1.5708
NO
GO TO 43
44
ISIDE=1
THETA=THETAO+DELTHB*(FLOAT(J)-2.0)
THESID=(90.0/57.2957795+DELTHT)-DELTHT*FLOAT(J)
GO TO 43
53
THETA=THESID+DELTHT*(FLOAT(J)-1.0)
43
CONTINUE


GO TO 2
YES
NO
THETA.GT.-0.0001.AND.THETA.LT.3.1415
GO TO 15
YLECL.GT.0.0
YA=0.0
RADIUS=SQRT(2.0*RNOSE*XB-XB*XB)
ZA=RADIUS
THETA.LT.0.0
ZA=-RADIUS
GO TO 19
15
ZA=RNOSE
ZA=-RNOSE
GO TO 17
THETA.GT.1.5708
GO TO 19
ITOP.EQ.1
YA=YLECL
16
CONTINUE
GO TO 18
XB.GE.RNOSE
C=(RNOSE-XB)*SIN(SWEEP)/COS(SWEEP)
RADIUS=SQRT(2.0*RNOSE*XB-XB*XB)
YA=RADIUS*SIN(THETA)
YA.GT.C
ZA=-RADIUS*COS(THETA)
18
19
A=RNOSE*(1.0-COS(SWEEP))
XB.GE.A
YLE=YLECL+(RNOSE/SIN(SWEEP))







SYMBOLS USED IN SUBROUTINE SLABD

A	R	U	DISTANCE TO TRANSITION BETWEEN NOSE AND LEADING EDGE	SLABD
AA	R	U	SEMI-SPAN DISTANCE PARAMETER	SLABD
BB	R	U	SEMI-SPAN DISTANCE PARAMETER	SLABD
BDCT2	R	U	DUMMY VARIABLE	SLABD
BDOT	R	U	DUMMY VARIABLE	SLABD
BDOT2	R	U	DUMMY VARIABLE	SLABD
BOTTC	R	U	BOTTOM THICKNESS CORRECTION PARAMETER	SLABD
C	R	U	SEMI-SPAN DISTANCE PARAMETER	SLABD
CARD	R	D	ARRAY FOR READING CARD	SLABD
CASE	I	C	CASE NUMBER	SLABD
DELTHB	R	U	BOTTOM SURFACE DELTA-THETA INCREMENT	SLABD
DELTHT	R	U	TOP SURFACE DELTA THETA INCREMENT	SLABD
DELZ	R	U	GEOMETRY DATA DISPLACEMENT PARAMETER IN Z-DIRECTION	SLABD
DUMMY	R	U	DUMMY VARIABLE	SLABD
ERROR	I	C	ERROR FLAG	SLABD
I	I	U	INDEX COUNTER	SLABD
IP	I	U	CARD PRINT POSITION FLAG	SLABD
IPRINT	I	U	PRINT FLAG	SLABD
IREW8	I	U	TAPE 8 REWIND FLAG	SLABD
ISIDE	I	U	GEOMETRY ANGULAR POSITION INDICATOR	SLABD
ITOC	I	U	THICKNESS CORRECTION TABLE FLAG	SLABD
ITOP	I	U	TOP GEOMETRY CONTROL FLAG	SLABD
ITYPE	I	U	CARD TYPE	SLABD
I8BSP	I	U	TAPE 8 BACKSPACE CONTROL FLAG	SLABD
J	I	U	ANGULAR LOOP INDEX	SLABD
LAST	I	U	SLAB DELTA OPTION TERMINATION FLAG	SLABD
LAST2	I	U	FLAG TO INDICATE LAST CARD OF T/C TABLE	SLABD
LAST3	I	U	FLAG TO INDICATE LAST CROSS-SECTION CARD	SLABD
LINE	I	U	OUTPUT LINE COUNTER	SLABD
M	I	U	COUNTER	SLABD
MODE	I	U	GEOMETRY MODE FLAG	SLABD
N	I	U	COUNTER	SLABD
NOSPAN	I	U	NUMBER OF ELEMENT DIVISIONS FOR TOP OR BOTTOM	SLABD
PAGE	I	C	PAGE NUMBER	SLABD
PSPAN	R	U	PER CENT SEMI-SPAN	SLABD
R	R	U	LOCAL RADIUS	SLABD

SYMBOLS USED IN SUBROUTINE SLABD

RADIUS	R	U	RADIUS	SLABD
RNOSE	R	U	RADIUS OF NOSE AND LEADING EDGE	SLABD
RONS	I	U	NOT USED	SLABD
SEQ	I	U	CARD SEQUENCE NUMBER	SLABD
STAT	I	U	COORDINATE POINT STATUS FLAG	SLABD
STATA	I	U	COORDINATE POINT STATUS FLAG	SLABD
STATT	I	U	COORDINATE POINT STATUS FLAG	SLABD
SWEEP	R	U	LEADING EDGE SWEEP ANGLE	SLABD
THCHK	R	U	ANGULAR CHECK PARAMETER	SLABD
THEMAX	R	U	MAXIMUM VALUE OF THETA	SLABD
THESID	R	U	THETA AT THE SIDE	SLABD
THETA	R	U	ANGULAR POSITION IN Z-Y PLANE (FROM BOTTOM)	SLABD
THETAB	I	U	NUMBER OF ANGULAR DIVISIONS ON THE BOTTOM	SLABD
THETAO	R	U	STARTING VALUE OF THETA	SLABD
THETAT	I	U	NUMBER OF ANGULAR DIVISIONS ON THE TOP	SLABD
THETA2	R	U	ANGULAR POSITION	SLABD
TITLE	R	C	TITLE	SLABD
TOPTC	R	U	TOP THICKNESS CORRECTION FACTOR	SLABD
TYPE	I	U	CARD TYPE NUMBER	SLABD
X	R	U	X-COORDINATE	SLABD
XA	R	U	X-COORDINATE	SLABD
XO	R	U	INPUT X-COORDINATE STATION	SLABD
XX	R	U	X-COORDINATE	SLABD
Y	R	U	Y-COORDINATE	SLABD
YA	R	U	Y-COORDINATE	SLABD
YLE	R	U	Y-DISTANCE TO THE LEADING EDGE	SLABD
YLECL	R	U	Y-DISTANCE TO LEADING EDGE CENTER LINE	SLABD
YY	R	U	Y-COORDINATE	SLABD
Z	R	U	Z-COORDINATE	SLABD
ZA	R	U	Z-COORDINATE	SLABD
ZB	R	D	Z-COORDINATE	SLABD
ZFACT	R	U	THICKNESS CORRECTION FACTOR FROM TABLES	SLABD
ZT	R	D	THICKNESS CORRECTION TABLE ARRAY	SLABD
ZZ	R	U	Z-COORDINATE	SLABD

```
SYMBOLS USED IN SUBROUTINE SLABD

RADIUS   R  U  RADIUS                                              SLABD
RNOSE    R  U  RADIUS OF NOSE AND LEADING EDGE                     SLABD
ROWS     I  U  NOT USED                                            SLABD
SEQ      I  U  CARD SEQUENCE NUMBER                                SLABD
STAT     I  U  COORDINATE POINT STATUS FLAG                        SLABD
STATA    I  U  COORDINATE POINT STATUS FLAG                        SLABD
STATT    I  U  COORDINATE POINT STATUS FLAG                        SLABD
SWEEP    R  U  LEADING EDGE SWEEP ANGLE                            SLABD
THCHK    R  U  ANGULAR CHECK PARAMETER                             SLABD
THEMAX   R  U  MAXIMUM VALUE OF THETA                              SLABD
THESID   R  U  THETA AT THE SIDE                                   SLABD
THETA    R  U  ANGULAR POSITION IN Z-Y PLANE (FROM BOTTOM)         SLABD
THETAB   I  U  NUMBER OF ANGULAR DIVISIONS ON THE BOTTOM           SLABD
THETAO   R  U  STARTING VALUE OF THETA                             SLABD
THETAT   I  U  NUMBER OF ANGULAR DIVISIONS ON THE TOP              SLABD
THETA2   R  U  ANGULAR POSITION                                    SLABD
TITLE    R  C  TITLE                                               SLABD
TOPTC    R  U  TOP THICKNESS CORRECTION FACTOR                     SLABD
TYPE     I  U  CARD TYPE NUMBER                                    SLABD
X        R  U  X-COORDINATE                                        SLABD
XA       R  U  X-COORDINATE                                        SLABD
XB       R  U  INPUT X-COORDINATE STATION                          SLABD
XX       R  U  X-COORDINATE                                        SLABD
Y        R  U  Y-COORDINATE                                        SLABD
YA       R  U  Y-COORDINATE                                        SLABD
YLE      R  U  Y-DISTANCE TO THE LEADING EDGE                      SLABD
YLECL    R  U  Y-DISTANCE TO LEADING EDGE CENTER LINE              SLABD
YY       R  U  Y-COORDINATE                                        SLABD
Z        R  U  Z-COORDINATE                                        SLABD
ZA       R  U  Z-COORDINATE                                        SLABD
ZB       R  D  Z-COORDINATE                                        SLABD
ZFACT    R  U  THICKNESS CORRECTION FACTOR FROM TABLES             SLABD
ZT       R  D  THICKNESS CORRECTION TABLE ARRAY                    SLABD
ZZ       R  U  Z-COORDINATE                                        SLABD
```

33. SUBROUTINE TTABLE (DECK SLBB)

This routine performs the interpolation to find the thickness correction factors for the Slab Delta Routine.

a. Algorithm

Search for the proper points in the data table to be used in the interpolation. Call on the quadratic interpolation routine, QINT, to obtain the interpolated value.

b. Input/Output

None

c. Error

None

d. Subroutines Required

QINT

e. Argument List

(A, B, C, D, R, G, GI)

f. Length

1888 bytes

DECK SLBB

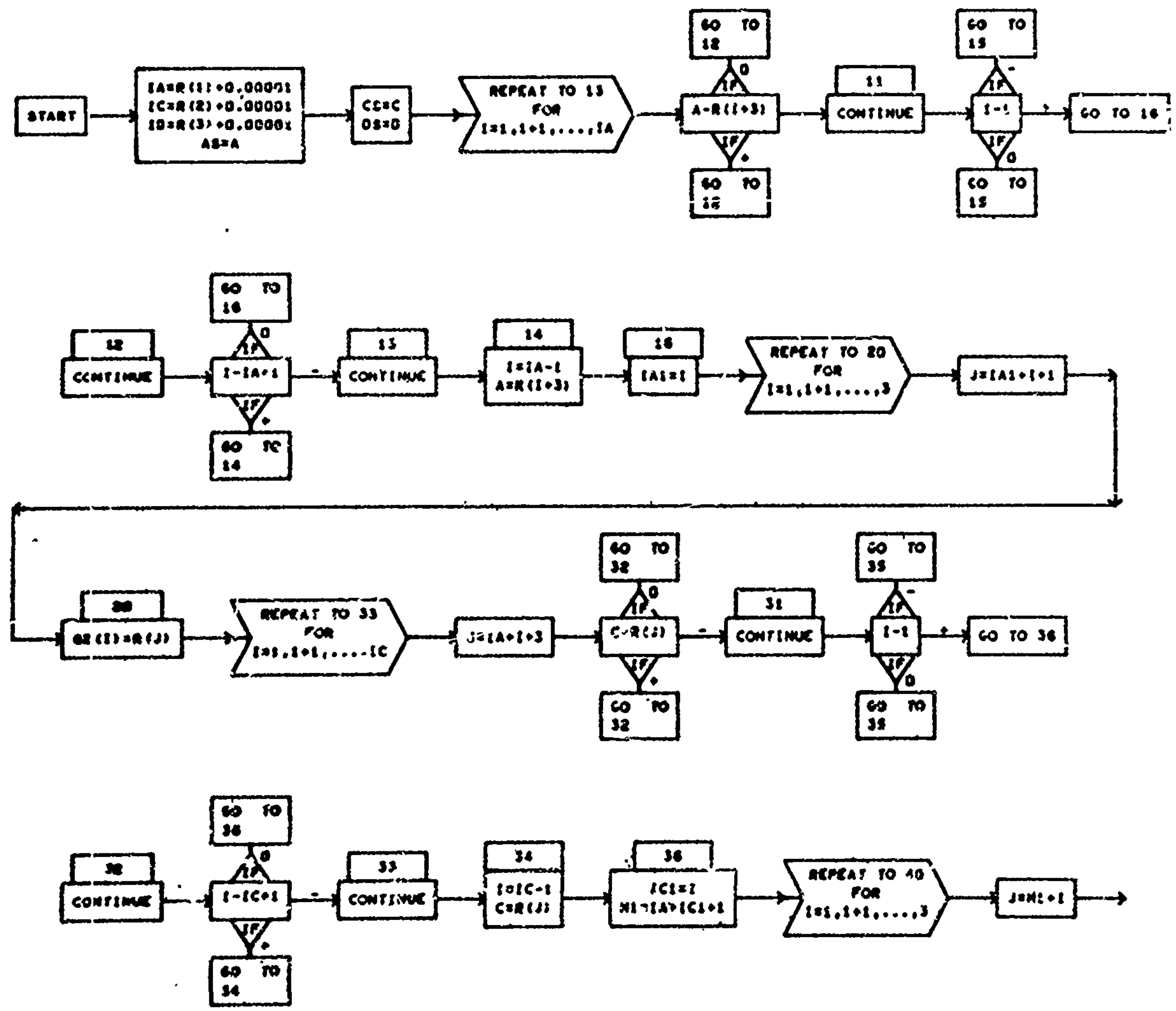
```
      SUBROUTINE TTABLE (A,B,C,D,R,G,G1)
C          TRIPLE  INTERPOLATION  ROUTINE
      DIMENSION R(300),Q1(3),Q2(3),Q3(3),Q8(3),Q10(3),Q11(3)
C     DIMENSION R(300),Q1(3),Q2(3),Q3(3),Q8(3),Q10(3),Q11(3)
          IA = R(1) + 0.00001
          IC = R(2) + 0.00001
          ID = R(3) + 0.00001
        AS = A
        CS = C
        DS = D
         DO 13 I = 1,IA
        IF(A-R(I+3))11,12,12
   11   IF(I-1) 15,15,16
   12   IF(I-IA+1) 13,16,14
   13 CONTINUE
   14     I = IA-1
          A = R(I+3)
   16     IA1 = I
         DO 20 I =1,3
          J = IA1+I+1
   20     Q2(I)= R(J)
C       C ARGUMENT CHECK
         DO 33 I = 1, IC
          J = IA+I+3
        IF(C-R(J)) 31,32,32
   31   IF(I-1) 35,35,36
   32   IF(I-IC+1) 33,36,34
   33 CONTINUE
   34     I = IC-1
          C = R(J)
   36     IC1 = I
C       C ARGUMENT ARRAY
          M1 = IA+IC1+1
         DO 40 I = 1,3
          J = M1+I
```

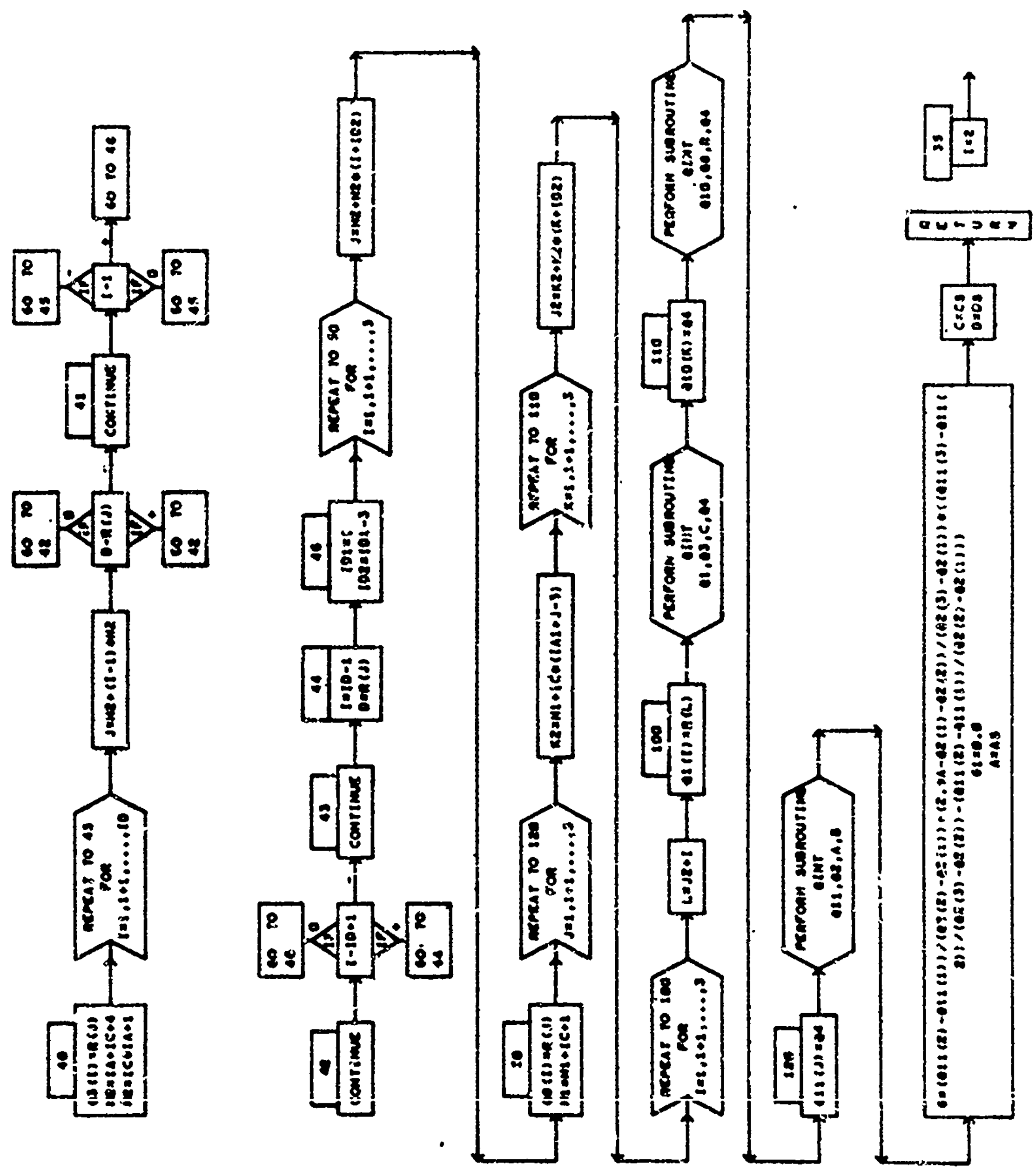
DECK SLBB

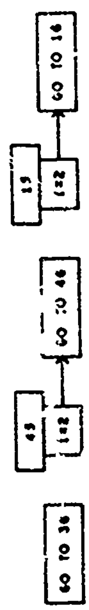
```
   40        Q3(I) = R(J)
C          D ARGUMENT CHECK
             M2 = IA+IC+4
             N2 = IC*IA+1
           DO 43 I = 1,ID
             J = M2+(I-1)*N2
          IF(D-R(J)) 41,42,42
   41     IF(I-1) 45,45,45
   42     IF(I-ID+1) 43,46,44
   43 CONTINUE
   44        I = ID-1
             D  = R(J)
   46        ID1 = I
C          D ARGUMENT ARRAY
             ID2 = ID1-3
        DO 50 I=1,3
             J = M2+N2*(I+ID2)
   50        Q8(I) = R(J)
             N1 = M1+IC+1
        DO 120 J=1,3
        K2 = N1 + IC*(IA1+J-3)
        DO 110 K=1,3
        J2 = K2 + N2*(K+ID2)
        DO 100 I=1,3
        L = J2 + I
  100 Q1(I) = R(L)
        CALL QINT (Q1,Q3,C,Q4)
  110 Q10(K) = Q4
        CALL QINT (Q10,Q8,D,Q4)
  120 Q11(J) = Q4
        CALL QINT (Q11,Q2,A,B)
        G = (Q11(2)-Q11(1))/(Q2(2)-Q2(1))+(2.*A-Q2(1)-Q2(2))/(Q2(3)-Q2(1))
       1  *((Q11(3)-Q11(2))/(Q2(3)-Q2(2))-(Q11(2)-Q11(1))/(Q2(2)-Q2(1)))
        G1= 0.0
         A = AS
         C = CS
```

```
DECK SLBB

      D = DS                                                  SLBB  0720
      RETURN                                                  SLBB  0730
   35   I = 2                                                 SLBB  0740
        GO TO 36                                              SLBB  0750
   45   I = 2                                                 SLBB  0760
      GO TO 46                                                SLBB  0770
   15   I = 2                                                 SLBB  0780
        GO TO 16                                              SLBB  0790
      END                                                     SLBB  0800
```

SUBROUTINE FTABLE






GO TO 16
I=2
GO TO 46
I=2
GO TO 36

SYMBOLS USED IN SUBROUTINE TTABLE

A	R	A	FIRST INDEPENDENT VARIABLE (PER CENT SEMI-SPAN)	TTABLE
AS	R	U	SAVED VALUE OF FIRST INDEPENDENT VARIABLE	TTABLE
B	R	A	DEPENDENT VARIABLE (ZFACT)	TTABLE
C	R	A	DUMMY VARIABLE	TTABLE
CS	R	U	SAVED VALUE OF DUMMY VARIABLE	TTABLE
D	R	A	SECOND INDEPENDENT VARIABLE (STATION)	TTABLE
DS	R	U	SAVED VALUE OF SECOND INDEPENDENT VARIABLE	TTABLE
G	R	A	FIRST DERIVATIVE (NOT USED)	TTABLE
G1	R	A	SECOND DERIVATIVE (NOT USED)	TTABLE
I	I	U	DO-LOOP INDEX	TTABLE
IA	I	U	INDEX	TTABLE
IA1	I	U	INDEX	TTABLE
IC	I	U	INDEX	TTABLE
IC1	I	U	INDEX	TTABLE
ID	I	U	INDEX	TTABLE
ID1	I	U	INDEX	TTABLE
ID2	I	U	INDEX	TTABLE
J	I	U	INDEX	TTABLE
J2	I	U	INDEX	TTABLE
K	I	U	DO-LOOP INDEX	TTABLE
K2	I	U	INDEX	TTABLE
L	I	U	INDEX	TTABLE
M1	I	U	INDEX	TTABLE
M2	I	U	INDEX	TTABLE
N1	I	U	INDEX	TTABLE
N2	I	U	INDEX	TTABLE
Q1	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q10	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q11	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q2	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q3	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q4	R	U	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q8	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
R	R	A	DATA ARRAY	TTABLE

34. SUBROUTINE QINT (DECK SLBC)

a. Algorithm

Perform a quadratic interpolation with the given values.

b. Input/Output

None

c. Error

None

d. Subroutines Required

None

e. Argument List

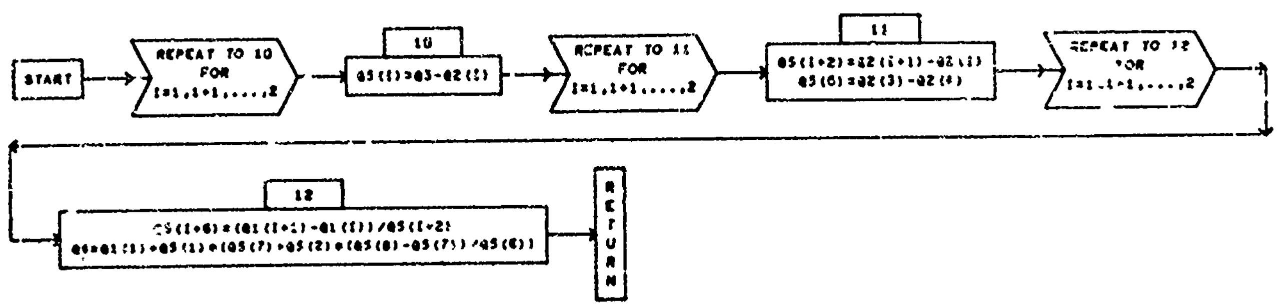
(Q1, Q2, Q3, Q4)

f. Length

490 bytes

```
DECK SLBC

      SUBROUTINE QINT(Q1,Q2,Q3,Q4)                                      SLBC  0010
      DIMENSION   Q1(3),Q2(3),Q5(8)                                     SLBC  0020
C     DIMENSION   Q1(3),Q2(3),Q5(8)                                     SLBC  0030
         DO 10 I = 1,2                                                  SLBC  0040
   10       Q5(I) = Q3 - Q2(I)                                          SLBC  0050
         DO 11 I = 1,2                                                  SLBC  0060
   11       Q5(I+2) = Q2(I+1) -Q2(I)                                    SLBC  0070
            Q5(6) = Q2(3) -Q2(1)                                        SLBC  0080
         DO 12 I = 1,2                                                  SLBC  0090
   12       Q5(I+6) =(Q1(I+1)-Q1(I))/Q5(I+2)                            SLBC  0100
            Q4 = Q1(1) + Q5(1)*(Q5(7)+Q5(2)*(Q5(8)-Q5(7))  /Q5(6)    )  SLBC  0110
      RETURN                                                            SLBC  0120
      END                                                               SLBC  0130
```

SUBROUTINE QINT



SYMBOLS USED IN SUBROUTINE QINT

Q1	R	A	QUADRATIC INTERPOLATION DATA ARRAY	QINT
Q2	R	A	QUADRATIC INTERPOLATION DATA ARRAY	QINT
Q3	R	A	QUADRATIC INTERPOLATION DATA ARRAY	QINT
Q4	R	A	QUADRATIC INTERPOLATION DATA ARRAY	QINT
Q5	R	D	QUADRATIC INTERPOLATION DATA ARRAY	QINT

35. SUBROUTINE CARD (DECK CARD)

This routine reads geometry data from a tape unit and punches the information on cards.

a. Algorithm

The version for use with the IBM 360 reads the data from Tape 8 and writes the same information on the punch unit (Unit 7).

b. Input/Output

Reads geometry data from Tape 8 and writes the same information on the punch unit (Unit 7)

c. Error

None

d. Subroutines Required

None

e. Argument List

None

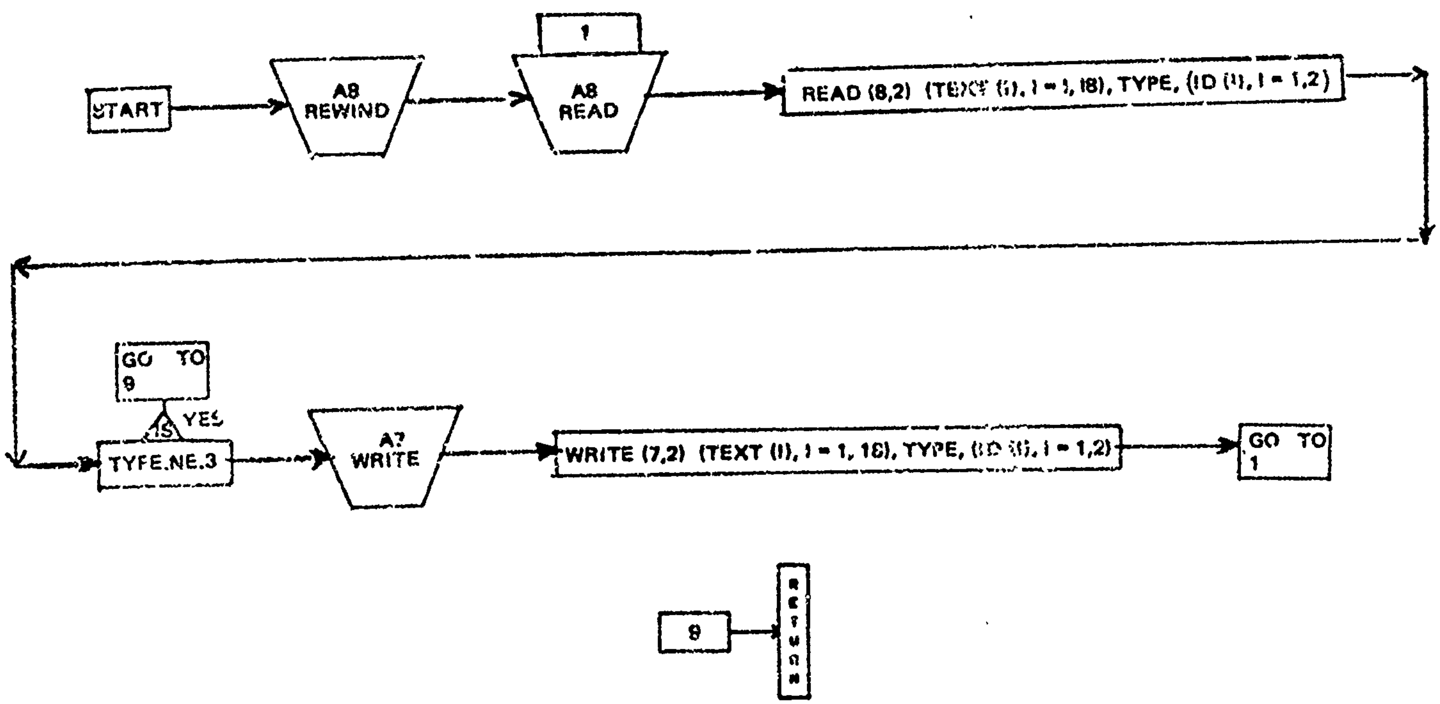
f. Length

460 bytes

DECK CARD

```
      SUBROUTINE CARD                                                   CARD  0010
C   IBM 360 VERSION OF SUBROUTINE CARD                                  CARD  0020
C   THIS SUBROUTINE (OPTION 5) READS DATA FROM TAPE 8 AND WRITES THE    CARD  0030
C   SAME INFORMATION ON THE PUNCH UNIT (UNIT 7).                        CARD  0040
C                                                                       CARD  0050
      DIMENSION  TEXT(18),ID(2)                                         CARD  0060
      INTEGER TYPE                                                      CARD  0070
C                                                                       CARD  0080
      REWIND 8                                                          CARD  0090
    1 READ (8,2) (TEXT(I),I=1,18),TYPE,(ID(I),I=1,2)                    CARD  0100
    2 FORMAT (17A4,1A3,I1,2A4)                                          CARD  0110
      IF (TYPE.NE.3) GO TO 9                                            CARD  0120
C                                                                       CARD  0130
      WRITE (7,2) (TEXT(I),I=1,18),TYPE,(ID(I),I=1,2)                   CARD  0140
      GO TO 1                                                           CARD  0150
C                                                                       CARD  0160
    9 RETURN                                                            CARD  0170
      END                                                               CARD  0180
```

SUBROUTINE CARD



CARD

```
SYMBOLS USED IN SUBROUTINE CARD

ID       I  D  INFORMATION ON TYPE 3 CARD FROM CC 73 TO CC 80          CARD
TEXT     I  D  INFORMATION ON TYPE 3 CARD FROM CC 1 TO CC 71           CARD
TYPE     I  U  CARD TYPE                                               CARD
```

APPENDIX B

PROGRAM MNEMONIC LIST

This appendix contains an alphabetic list of all symbols used in the Mark III Mod 0 program. The list is divided into five fields which are described as follows:

(i) The first field contains the symbol

(ii) The second field contains the letters I, L, or R, denoting integer, logical, or real variable respectively.

(iii) The third field contains the letters A, C, D, U, denoting argument list, common, dimensioned, or undimensioned, variable respectively. The hierarchy of the above letters is A, C, D, U.

(iv) The fourth field contains the definition of the symbol.

(v) The fifth field contains the name of the subroutine in which the symbol occurs.

SYMBOL	TYPE		DEFINITION	ROUTINE
A	R	U	ELLIPSE RADIUS IN Y-DIRECTION	ANALY1
A	R	D	MATRIX OF ATMOSPHERIC PROPERTIES	ATMOS
A	D	D	ELLIPTICAL INTEGRAL PARAMETER	ELP1
A	R	D	ITERATION VARIABLE ARRAY	EXPAND
A	R	U	SPEED OF SOUND	NEWTPM
A	R	D	FIRST DATA ARRAY	PLOT
A	R	A	POLYNOMIAL COEFFICIENT ARRAY	POLY
A	R	U	COEFFICIENT USED IN SPALDING-CHI METHOD	QC
A	R	U	SPEED OF SOUND	SKINFR
A	R	U	DISTANCE TO TRANSITION BETWEEN NOSE AND LEADING EDGE	SLABD
A	R	A	FIRST INDEPENDENT VARIABLE (PER CENT SEMI-SPAN)	TTABLE
AA	R	D	Y-DIRECTION ELLIPSE RADIUS ARRAY	ANALY1
AA	R	D	SEVENTH DATA ARRAY	PLOT
AA	R	U	SEMI-SPAN DISTANCE PARAMETER	SLABD
AFLAG	L	U	INPUT DATA READ CONTROL FLAG	PICTUR
AFLAG	L	U	INPUT DATA READ CONTROL FLAG	SDATA
AFS	R	U	FREE-STREAM SPEED OF SOUND	FORCE
AFS	R	A	FREE-STREAM SPEED OF SOUND	SHKEXP
AFS	R	A	FREE-STREAM SPEED OF SOUND	SKINFR
AK	R	A	THE MODULUS	ELP1
AKD	D	U	MODULUS	ELP1
AKK	R	A	ELLIPTICAL INTEGRAL OF THE FIRST KIND	ELP1
ALP	R	C	ANGLE OF ATTACK ARRAY	AERO
ALP	R	C	ANGLE OF ATTACK ARRAY	FORCE
ALP	R	C	ANGLE OF ATTACK ARRAY	SKINFR
ALP	R	C	ANGLE OF ATTACK ARRAY	TEMP
ALPHA	R	U	ANGLE OF ATTACK, DEGREES	AERO
ALPHA	R	A	ANGLE OF ATTACK, DEGREES	FORCE
ALPHA	R	A	ANGLE OF ATTACK, DEGREES	SKINFR
ALPHA	R	A	ANGLE OF ATTACK, DEGREES	VECTOR
ALPHAR	R	U	ANGLE OF ATTACK, RADIANS	FORCE
ALPHAR	R	U	ANGLE OF ATTACK, RADIANS	VECTOR
ALPHAS	R	U	SAVED VALUE OF ANGLE OF ATTACK	AERO
ALT	R	U	ALTITUDE, FEET	AERO
ALT	R	A	ALTITUDE, FEET	FORCE
ALT	R	A	ALTITUDE, FEET	SKINFR

SYMBOL	TYPE		DEFINITION	ROUTINE
ANGLE	R	A	FLOW ANGLE ARRAY	COMPR
ANGLE	R	A	FLOW ANGLE ARRAY	CONE
ANGLE	R	A	FLOW ANGLE ARRAY	DELWNG
ANGLE	R	A	FLOW ANGLE ARRAY	EXPAND
ANGLE	R	D	FLOW ANGLE ARRAY	FLOSEP
ANGLE	R	D	FLOW ANGLE ARRAY	FORCE
ANGLE	R	A	FLOW ANGLE ARRAY	NEWTPM
ANGLE	R	D	FLOW ANGLE ARRAY	SHKEXF
ANGLE	R	D	FLOW ANGLE ARRAY	SKINFR
AOD	R	U	COEFFICIENT IN DEFINITION OF ODD ORIGIN	BLUNT
AR	R	U	ASPECT RATIO OF WING/TAIL	PLUNGE
AREA	R	U	ELEMENT AREA	CONTRL
AREA	R	A	ELEMENT AREA	FLOSEP
AREA	R	U	ELEMENT AREA	FORCE
AREA	R	U	ELEMENT AREA	PICTUR
AREA	R	U	ELEMENT AREA	SDATA
AREAS	R	U	TOTAL AREA OF A COLUMN OF ELEMENTS	FORCE
AREAT	R	U	TOTAL AREA	FORCE
AREAT	R	U	TOTAL AREA	PICTUR
AREAT	R	U	TOTAL AREA	SDATA
AREA2	R	C	SURFACE AREA OF QUADRILATERALS	AERO
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	COMPR
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	CONE
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	CONTRL
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	DELWNG
AREA2	R	C	QNADRILATERAL ELEMENT AREA ARRAY	EXPAND
AREA2	R	C	(NOT USED)	FLOSEP
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	FORCE
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	NEWTPM
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	QC
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	SDATA
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	SHKEXP
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	SKINFR
AREA2	R	C	QUADRILATERAL ELEMENT AREA ARRAY	TEMP
AS	R	U	SAVED VALUE OF FIRST INDEPENDENT VARIABLE	TTABLE
ATA	D	U	ELLIPTICAL INTEGRAL PARAMETER	ELP1

SYMBOL	TYPE		DEFINITION	ROUTINE
AVX	R	U	AVERAGE POINT COORDINATE-X	PICTUR
AVX	R	U	AVERAGE POINT COORDINATE-X	SDATA
AVY	R	U	AVERAGE POINT COORDINATE-Y	PICTUR
AVY	R	U	AVERAGE POINT COORDINATE-Y	SDATA
AVZ	R	U	AVERAGE POINT COORDINATE-Z	PICTUR
AVZ	R	U	AVERAGE POINT COORDINATE-Z	SDATA
AX	R	D	X-STATION ARRAY	ANALY1
A1	R	A	ATMOSPHERIC SPEED OF SOUND, FEET/SECOND	ATMOS
A1	R	U	COEFFICIENT IN DEFINITION OF FIRST FUNCTION	BLUNT
A1	R	U	ROTATION MATRIX CONSTANT	PICTUR
A1	R	U	COEFFICENT IN THE DEFINITION OF REFERENCE CONDITION	QC
A2	R	U	ROTATION MATRIX CONSTANT	PICTUR
A2	R	U	COEFFICIENT IN THE DEFINITION OF REFERENCE CONDITION	QC
A3	R	A	GEOMETRIC ALTITUDE, FEET	ATMOS
A3	R	U	ROTATION MATRIX CONSTANT	PICTUR
A4	R	A	ATMOSPHERIC PRESSURE, POUNDS PER SQUARE FOOT	ATMOS
A4	R	U	ROTATION MATRIX CONSTANT	PICTUR
A5	R	U	ROTATION MATRIX CONSTANT	PICTUR
A6	R	A	ATMOSPHERIC DENSITY, SLUGS PER CUBIC FOOT	ATMOS
A6	R	U	ROTATION MATRIX CONSTANT	PICTUR
A7	R	U	ROTATION MATRIX CONSTANT	PICTUR
A8	R	A	ATMOSPHERIC TEMPERATURE, DEGREE RANKINE	ATMOS
A8	R	U	ROTATION MATRIX CONSTANT	PICTUR
A9	R	U	ROTATION MATRIX CONSTANT	PICTUR
B	R	U	ELLIPSE RADIUS IN Z-DIRECTION	ANALY1
B	R	U	VARIABLE IN CUBIC EQUATION	COMPR
B	D	D	ELLIPTICAL INTEGRAL CONSTANT ARRAY	ELP1
B	R	D	SECOND DATA ARRAY	PLOT
B	R	U	WING/TAIL SPAN	PLUNGE
B	R	U	COEFFICIENT USED IN SPALDING-CHI METHOD	QC
B	R	U	LAMINAR VISCOUS INTERACTION PARAMETER	SKINFR
B	R	A	DEPENDENT VARIABLE (ZFACT)	TTABLE
BA	R	U	PRODUCT OF BETA AND ASPECT RATIO	PLUNGE
BARS	R	U	PRODUCT OF BA AND RS	PLUNGE
BA4	R	U	BA DIVIDED BY 4	PLUNGE
BB	R	D	Z-DIRECTION ELLIPSE RADIUS ARRAY	ANALY1

SYMBOL	TYPE	DEFINITION	ROUTINE
BB	R D	EIGHT DATA ARRAY	PLOT
BB	R U	SEMI-SPAN DISTANCE PARAMETER	SLABD
BDCR	R U	PRODUCT OF BETA AND D DIVIDED BY CR	PLUNGE
BDCT2	R U	DUMMY VARIABLE	SLABD
BDOT	R U	DUMMY VARIABLE	SLABD
BDOT2	R U	DUMMY VARIABLE	SLABD
BET	R C	YAW ANGLE ARRAY	AERO
BET	R C	YAW ANGLE ARRAY	FORCE
BET	R C	YAW ANGLE ARRAY	SKINFR
BET	R C	YAW ANGLE ARRAY	TEMP
BETA	R U	YAW ANGLE, DEGREES	AERO
BETA	R A	YAW ANGLE, DEGREES	FORCE
BETA	R U	PRANDTL-GLAUERT FACTOR	PLUNGE
BETA	R A	YAW ANGLE, DEGREES	VECTOR
BETAR	R U	YAW ANGLE, RADIANS	FORCE
BETAR	R U	YAW ANGLE, RADIANS	VECTOR
BETAS	R U	SAVED VALUE OF YAW ANGLE	AERO
BFLAG	L U	INPUT DATA READ CONTROL FLAG	PICTUR
BFLAG	L U	INPUT DATA READ CONTROL FLAG	SDATA
BM	R U	PRODUCT OF BETA AND M	PLUNGE
BMR2	R U	SQUARE ROOT OF DIFFERENCE OF BM2 AND 1.0	PLUNGE
BM1	R U	PRODUCT OF BETA AND M	PLUNGE
BM2	R U	SQUARE OF PRODUCT OF BETA AND M	PLUNGE
BOD	R U	COEFFICIENT IN DEFINITION OF ODD ORIGIN	BLUNT
BOTTC	R U	BOTTOM THICKNESS CORRECTION PARAMETER	SLABD
BS	R C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	AERO
BS	R C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	COMPR
BS	R C	FLOW CONDITIONS BEHIND COMPRESSION	CONE
BS	R C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	DELWNG
BS	R C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	EXPAND
BS	R C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	FLOSEP
BS	R C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	FORCE
BS	R C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	NEWTPM
BS	R C	FLOW CONDITION ARRAY BEHIND SHOCK OR EXPANSION	QC
BS	R C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	SHKEXP
BS	R C	FLOW CONDITIONS BEHIND SHOCK OR EXPANSION	SKINFR

SYMBOL	TYPE		DEFINITION	ROUTINE
BS	R	C	FLOW CONDITIONS BEHIND THE SHOCK OR EXPANSION	TEMP
BS2	R	U	PRESSURE TO BE SAVED	FLOSEP
BS2F	R	U	INVISCID SHOCK-EXPANSION PRESSURE	FLOSEP
BS3	R	U	TEMPERATURE TO BE SAVED	FLOSEP
BS3F	R	U	TEMPERATURE TO BE SAVED	FLOSEP
BS6	R	U	MACH NUMBER TO BE SAVED	FLOSEP
BS6F	R	U	MACH NUMBER ON FLAP TO BE SAVED	FLOSEP
BS6SQ	R	U	MACH NUMBER SQUARED	COMPR
BS6SQ	R	U	MACH NUMBER SQUARED	CONE
BS6SQ	R	U	MACH NUMBER SQUARED	DELWNG
B1	R	U	COEFFICIENT IN DEFINITION OF FIRST FUNCTION	BLUNT
C	R	U	VARIABLE IN CUBIC EQUATION	COMPR
C	D	U	ELLIPTICAL INTEGRAL PARAMETER	ELP1
C	R	D	ITERATION VARIABLE ARRAY	EXPAND
C	R	D	THIRD DATA ARRAY	PLOT
C	R	U	REFERENCE CHORD FOR BODY	PLUNGE
C	R	U	COEFFICIENT USED IN SPALDING-CHI METHOD	QC
C	R	U	SEMI-SPAN DISTANCE PARAMETER	SLABD
C	R	A	DUMMY VARIABLE	TTABLE
CA	R	U	AXIAL FORCE COEFFICIENT	FORCE
CA	R	A	AXIAL FORCE COEFFICIENT	SKINF
CA	R	A	AXIAL FORCE COEFFICIENT	VECTOR
CAI	R	U	AXIAL FORCE INCREMENT	AERO
CAI	R	A	AXIAL FORCE INCREMENT	FORCE
CARD	R	D	ARRAY FOR READING IN 80 COLUMN CARD	AERO
CARD	R	D	ARRAY FOR READING IN 80 COLUMN CARD	PICTUR
CARD	R	D	ARRAY FOR READING CARD	SLABD
CASD1	R	U	FIRST VALUE OF AXIAL FORCE FOR DERIVATIVE CALCULATIONS	AERO
CASD2	R	U	SECOND VALUE OF AXIAL FORCE FOR DERIVATIVE CALCULATIONS	AERO
CASE	I	C	CASE NUMBER	AERO
CASE	I	C	CASE NUMBER	ANALY1
CASE	I	C	CASE NUMBER	ANALY2
CASE	I	C	CASE NUMBER	COMPR
CASE	I	C	CASE NUMBER	CONE
CASE	I	C	CASE NUMBER	CONTRL
CASE	I	C	CASE NUMBER	DELWNG

SYMBOL	TYPE	DEFINITION	ROUTINE
CASE	I C	CASE NUMBER	EXPAND
CASE	I C	CASE NUMBER	FLOSEP
CASE	I C	CASE NUMBER	FORCE
CASE	I C	CASE NUMBER	GRAPIC
CASE	I C	CASE NUMBER	HEADER
CASE	I C	CASE NUMBER	HEADR2
CASE	I C	PROBLEM CASE NUMBER	MAIN
CASE	I C	CASE NUMBER	NEWTPM
CASE	I C	CASE NUMBER	PICTUR
CASE	I C	CARD NUMBER	PLOT
CASE	I C	CASE NUMBER	PLUNGE
CASE	R C	CASE NUMBER	PUNCH
CASE	I C	CASE NUMBER	QC
CASE	I C	CASE NUMBER	SDATA
CASE	I C	CASE NUMBER	SHKEXP
CASE	I C	CASE NUMBER	SKINFR
CASE	I C	CASE NUMBER	SLABO
CASE	I C	CASE NUMBER	TEMP
CASE	I C	CASE NUMBER	VECTOR
CC	R D	NINTH DATA ARRAY	PLOT
CCA	R C	AXIAL FORCE COEFFICIENT ARRAY	AERO
CCA	R C	AXIAL FORCE COEFFICIENT ARRAY	FORCE
CCA	R C	AXIAL FORCE COEFFICIENT ARRAY	SKINFR
CCA	R C	AXIAL FORCE COEFFICIENT ARRAY	TEMP
CCAS	R D	SAVED VALUES OF AXIAL FORCE COEFFICIENT	AERO
CCD	R C	DRAG COEFFICIENT ARRAY	AERO
CCD	R C	DRAG COEFFICIENT ARRAY	FORCE
CCD	R C	DRAG COEFFICIENT ARRAY	SKINFR
CCD	R C	DRAG COEFFICIENT ARRAY	TEMP
CCDS	R D	SAVED VALUES OF DRAG COEFFICIENT	AERO
CCL	R C	LIFT COEFFICIENT ARRAY	AERO
CCL	R C	LIFT COEFFICIENT ARRAY	FORCE
CCL	R C	LIFT COEFFICIENT ARRAY	SKINFR
CCL	R C	LIFT COEFFICIENT ARRAY	TEMP
CCLL	R C	ROLLING MOMENT COEFFICIENT ARRAY	AERO
CCLL	R C	ROLLING MOMENT COEFFICIENT ARRAY	FORCE

SYMBOL	TYPE		DEFINITION	ROUTINE
CCLL	R	C	ROLLING MOMENT COEFFICIENT ARRAY	SKINFR
CCLL	R	C	ROLLING MOMENT COEFFICIENT ARRAY	TEMP
CCLLS	R	D	SAVED VALUES OF ROLLING MOMENT COEFFICIENT	AERO
CCLM	R	C	PITCHING MOMENT COEFFICIENT ARRAY	AERO
CCLM	R	C	PITCHING MOMENT COEFFICIENT ARRAY	FORCE
CCLM	R	C	PITCHING MOMENT COEFFICIENT ARRAY	SKINFR
CCLM	R	C	PITCHING MOMENT COEFFICIENT ARRAY	TEMP
CCLMS	R	D	SAVED VALUES OF PITCHING MOMENT COEFFICIENT	AERO
CCLN	R	C	YAWING MOMENT COEFFICIENT ARRAY	AERO
CCLN	R	C	YAWING MOMENT COEFFICIENT ARRAY	FORCE
CCLN	R	C	YAWING MOMENT COEFFICIENT ARRAY	SKINFR
CCLN	R	C	YAWING MOMENT COEFFICIENT ARRAY	TEMP
CCLNS	R	D	SAVED VALUES OF YAWING MOMENT COEFFICIENT	AERO
CCLS	R	D	SAVED VALUES OF LIFT COEFFICIENT	AERO
CCN	R	C	NORMAL FORCE COEFFICIENT ARRAY	AERO
CCN	R	C	NORMAL FORCE COEFFICIENT ARRAY	FORCE
CCN	R	C	NORMAL FORCE COEFFICIENT ARRAY	SKINFR
CCN	R	C	NORMAL FORCE COEFFICIENT ARRAY	TEMP
CCNS	R	D	SAVED VALUES OF NORMAL FORCE COEFFICIENT	AERO
CCY	R	C	SIDE FORCE COEFFICIENT ARRAY	AERO
CCY	R	C	SIDE FORCE COEFFICIENT ARRAY	FORCE
CCY	R	C	SIDE FORCE COEFFICIENT ARRAY	SKINFR
CCY	R	C	SIDE FORCE COEFFICIENT ARRAY	TEMP
CCYS	R	D	SAVED VALUES OF SIDE FORCE COEFFICIENT	AERO
CD	R	U	DRAG COEFFICIENT	FORCE
CD	R	A	DRAG COEFFICIENT	VECTOR
CDL	R	U	LAMINAR FLOW DRAG COEFFICIENT	SKINFR
CDT	R	U	TURBULENT FLOW DRAG COEFFICIENT	SKINFR
CF	R	C	SKIN FRICTION TOTAL AXIAL FORCE CONTRIBUTION	AERO
CF	R	C	SKIN FRICTION CONTRIBUTION TO AXIAL FORCE ARRAY	FORCE
CF	R	C	SKIN FRICTION CONTRIBUTION TO AXIAL FORCE ARRAY	SKINFR
CF	R	C	SKIN FRICTION CONTRIBUTION TO AXIAL FORCE ARRAY	TEMP
CFCFOL	R	U	LAMINAR RATIO OF SKIN FRACTION, INTERACTION/NO INTERACTION	SKINFR
CFCFOT	R	U	TURBULENT RATIO OF SKIN FRICTION = 1.0	SKINFR
CFCFO1	R	U	INITIAL SURFACE SKIN FRICTION RATIO	SKINFR
CFD1	R	U	FLAP CHORD/UPSTREAM INTERACTION LENGTH	FLOSEP

SYMBOL	TYPE		DEFINITION	ROUTINE
CFL	R	D	LAMINAR SKIN FRICTION COEFFICIENTS	SKINFR
CFLAP	R	U	FLAP CHORD	FLOSEP
CFLLOC	R	U	LOCAL LAMINAR SKIN-FRICTION COEFFICIENT	TEMP
CFO	R	U	SKIN-FRICTION COEFFICIENT WITHOUT INTERACTION	BLUNT
CFS	R	D	SAVED VALUES OF SKIN FRICTION TOTAL AXIAL FORCE CONTRIBUTION	AERO
CFT	R	D	TURBULENT SKIN FRICTION COEFFICIENTS	SKINFR
CFTLOC	R	C	LOCAL TURBULENT SKIN-FRICTION COEFFICIENT	QC
CFTLOC	R	C	LOCAL TURBULENT SKIN FRICTION COEFFICIENT	TEMP
CF1	R	U	SKIN FRICTION COEFFICIENT REFERENCED TO FREE-STREAM	TEMP
CF1RE1	R	U	SKIN FRICTION PARAMETER	TEMP
CHIBAR	R	U	HYPERSONIC INTERACTION PARAMETER	SKINFR
CKU	R	C	LAMINAR FLOW FLIGHT CONDITION CONSTANT	QC
CKU	R	C	LAMINAR FLOW FLIGHT CONDITION CONSTANT	SKINFR
CKU	R	C	LAMINAR FLOW FLIGHT CONDITION CONSTANT	TEMP
CL	R	U	LIFT COEFFICIENT ARRAY	FORCE
CL	R	A	LIFT COEFFICIENT	VECTOR
CLALW	R	U	LIFT-CURVE SLOPE FOR WING/TAIL (PER. RADIAN)	PLUNGE
CLAM	R	U	CHAPMAN-RUBESIN VISCOSITY COEFFICIENT, LAMINAR	SKINFR
CLL	R	U	ROLLING MOMENT COEFFICIENT	FORCE
CLL	R	A	ROLLING MOMENT COEFFICIENT	VECTOR
CLLI	R	U	ROLLING MOMENT COEFFICIENT INCREMENT	AERO
CLLI	R	A	ROLLING MOMENT COEFFICIENT INCREMENT	FORCE
CLLSD1	R	U	FIRST VALUE OF ROLLING MOMENT FOR DERIVATIVE	AERO
CLLSD2	R	U	SECOND VALUE OF ROLLING MOMENT FOR DERIVATIVE	AERO
CLM	R	U	PITCHING MOMENT COEFFICIENT	FORCE
CLM	R	A	PITCHING MOMENT COEFFICIENT	VECTOR
CLMI	R	U	PITCHING MOMENT COEFFICIENT INCREMENT	AERO
CLMI	R	A	PITCHING MOMENT COEFFICIENT INCREMENT	FORCE
CLMSD1	R	U	FIRST VALUE OF PITCHING MOMENT FOR DERIVATIVE	AERO
CLMSD2	R	U	SECOND VALUE OF PITCHING MOMENT FOR DERIVATIVE	AERO
CLN	R	U	YAWING MOMENT COEFFICIENT	FORCE
CLN	R	A	YAWING MOMENT COEFFICIENT	VECTOR
CLNI	R	U	YAWING MOMENT COEFFICIENT INCREMENT	AERO
CLNI	R	A	YAWING MOMENT COEFFICIENT MOMENT	FORCE
CLNSD1	R	U	FIRST VALUE OF YAWING MOMENT FOR DERIVATIVE	AERO
CLNSD2	R	U	SECOND VALUE OF YAWING MOMENT FOR DERIVATIVE	AERO

SYMBOL	TYPE	DEFINITION	ROUTINE
CLOD	R C	LIFT TO DRAG RATIO ARRAY	AERO
CLOD	R C	LIFT TO DRAG RATIO ARRAY	FORCE
CLOD	R C	LIFT TO DRAG RATIO ARRAY	SKINFR
CLOD	R C	LIFT TO DRAG RATIO ARRAY	TEMP
CLSD1	R U	FIRST VALUE OF LIFT COEFFICIENT FOR DERIVATIVE	AERO
CMA	R A	DERIVATIVE OF PITCHING MOMENT WITH ANGLE OF ATTACK	PLUNGE
CMADT	R A	PITCHING MOMENT-ALPHA DOT DERIVATIVE	PLUNGE
CMWPR	R U	WING-ALONE/TAIL-ALONE PITCHING MOMENT DERIVATIVE	PLUNGE
CN	R U	NORMAL FORCE COEFFICIENT	FORCE
CN	R A	NORMAL FORCE COEFFICIENT	SKINFR
CN	R A	NORMAL FORCE COEFFICIENT	VECTOR
CNI	R U	NORMAL FORCE COEFFICIENT INCREMENT	AERO
CNI	R A	NORMAL FORCE COEFFICIENT INCREMENT	FORCE
CNSD1	R U	FIRST VALUE OF NORMAL FORCE FOR DERIVATIVE	AERO
CNSD2	R U	SECOND VALUE OF NORMAL FORCE FOR DERIVATIVE	AERO
COEF	R U	COEFFICIENT DERIVATIVE	PLUNGE
COSDE	R U	COSINE OF CONTROL DEFLECTION ANGLE	CONTRL
COSDEL	R U	COSINE OF ANGLE NORMAL AND VELOCITY VECTORS	FORCE
COSEP1	R U	COSINE OF EPSILON 1	ANALY2
COSEP2	R U	COSINE OF EPSILON 2	ANALY2
COSPHI	R U	COSINE OF TRANSFORMATION ANGLE, PHI	CONTRL
COSPHI	R U	COSINE OF PHI	PICTUR
COSPSI	R U	COSINE OF TRANSFORMATION ANGLE, PSI	CONTRL
COSPSI	R U	COSINE OF PSI	PICTUR
COSTH	R U	COSINE OF THETA	PICTUR
CP	R A	PRESSURE COEFFICIENT	COMPR
CP	R U	PRESSURE COEFFICIENT	CONE
CP	R A	PRESSURE COEFFICIENT	CONE
CP	R A	PRESSURE COEFFICIENT	EXPAND
CP	R A	PRESSURE COEFFICIENT	FLOSEP
CP	R U	PRESSURE COEFFICIENT	FORCE
CP	R A	PRESSURE COEFFICIENT	NEWTPM
CP	R A	PRESSURE COEFFICIENT	SHKEXP
CP	R U	PRESSURE COEFFICIENT	SKINFR
CPAINC	R U	PRESSURE RISE TO CAUSE INCIPIENT SEPARATION	FLOSEP

SYMBOL	TYPE		DEFINITION	ROUTINE
CPAP	R	U	PLATEAU PRESSURE COEFFICIENT	FLOSEP
CPAVG	R	U	AVERAGE PRESSURE COEFFICIENT TIMES AREA FOR EQUIVALENT CONE	FORCE
CPA2IN	R	U	INVISCID PRESSURE RISE COEFFICIENT ON TO CONTROL SURFACE	FLOSEP
CPF	R	U	PRESSURE COEFFICIENT ON FLAP (DUMMY-NOT USED)	FLOSEP
CPIP	R	U	PLATEAU PRESSURE COEFFICIENT	FLOSEP
CPI2	R	U	PEAK PRESSURE COEFFICIENT ON FLAP	FLOSEP
CPNIN	R	A	PRESSURE COEFFICIENT (INPUT FORCE METHOD, INVISCID)	FLOSEP
CPNIN	R	U	MINIMUN PRESSURE COEFFICIENT	FORCE
CPQ	R	U	PRESSURE COEFFICIENT AT MATCHING POINT	NEWTPM
CPS	R	C	ARRAY FOR NEWTONIAN CORRELATION FACTOR, K	AERO
CPS	R	C	ARRAY FOR NEWTONIAN CORRELATION FACTOR, K	FORCE
CPS	R	C	ARRAY FOR NEWTONIAN CORRELATION FACTOR, K	SKINFR
CPS	R	C	ARRAY FOR NEWTONIAN CORRECTION FACTOR, K	TEMP
CPSEP	R	U	VISCOUS PRESSURE COEFFICIENT WITH SEPARATION	FLOSEP
CPSTAG	R	U	MODIFIED NEWTONIAN CORRELATION FACTOR, K	AERO
CPSTAG	R	A	MODIFIED NEWTONIAN CORRELATION FACTOR, K	COMPR
CPSTAG	R	A	NEWTONIAN CORRELATION FACTOR, K (STAGNATION PRESSURE COEFF)	FLOSEP
CPSTAG	R	A	MODIFIED NEWTONIAN CORRELATION FACTOR, K	FORCE
CPSTAG	R	A	MODIFIED NEWTONIAN CORRELATION FACTOR, K	NEWTPM
CPSTAG	R	A	MODIFIED NEWTONIAN CORRELATION FACTOR, K	SHKEXP
CPSTAG	R	A	MODIFIED NEWTONIAN CORRELATION FACTOR, K	SKINFR
CPX	R	U	PRESSURE COEFFICIENT AT SEPARATION POINT	FLOSEP
CP1	R	U	ITERATION PRESSURE COEFFICIENT	FORCE
CR	R	U	WING/TAIL CHORD AT WING/TAIL-BODY JUNCTURE	PLUNGE
CRBD	R	U	RICIPROCAL OF BDCR	PLUNGE
CRBD2	R	U	SQUARE OF CRBD	PLUNGE
CS	R	U	SAVED VALUE OF DUMMY VARIABLE	TTABLE
CSTAR	R	U	LINEAR VISCOSITY COEFFICIENT AT REFERENCE CONDITION	SKINFR
CTURB	R	U	CHAPMAN-RUBESIN VISCOSITY COEFFICIENT, TURBULENT	SKINFR
CWBYC	R	U	MEAN AERODYNAMIC CHORD OF EXPOSED WING/TAIL	PLUNGE
CY	R	U	SIDE FORCE COEFFICIENT	FORCE
CY	R	A	SIDE FORCE COEFFICIENT	VECTOR
CYB	R	[illegible]	DERIVATIVE OF SIDE FORCE WITH YAW ANGLE	PLUNGE
CYBDT	R	A	DERIVATIVE OF SIDE FORCE WITH BETA DOT	PLUNGE
CYI	R	U	SIDE FORCE COEFFICIENT INCREMENT	AERO
CYI	R	A	SIDE FORCE COEFFICIENT INCREMENT	FORCE

SYMBOL	TYPE	DEFINITION	ROUTINE
CYPRIM	R U	SIDE FORCE COEFFICIENT-WIND AXIS	FORCE
CYPRIM	R U	SIDE FORCE COEFFICIENT (WIND AXIS)	VECTOR
CYSD1	R U	FIRST VALUE OF SIDE FORCE COEFFICIENT FOR DERIVATIVE	AERO
CYSD2	R U	SECOND VALUE OF SIDE FORCE COEFFICIENT FOR DERIVATIVE	AERO
D	R D	CONSTANTS FOR BOUNDARY CURVES	ANALY2
D	R U	VARIABLE IN CUBIC EQUATION	COMPR
D	D U	ELLIPTICAL INTEGRAL PARAMETER	ELP1
D	R U	CORNER POINT PROJECTION DISTANCE	PICTUR
D	R D	FOURTH DATA ARRAY	PLOT
D	R U	BODY DIAMETER AT WING OR TAIL	PLUNGE
D	R U	CORNER POINT PROJECTION DISTANCE	SDATA
D	R A	SECOND INDEPENDENT VARIABLE (STATION)	TTABLE
DA	R U	ITERATION INCREMENT	EXPAND
DADC	R U	EXPANSION ITERATION PARAMETER	EXPAND
DCAA	R D	DERIVATIVE OF AXIAL FORCE WITH ANGLE OF ATTACK	AERO
DCAAS	R D	SAVED VALUES OF AXIAL FORCE-ANGLE OF ATTACK DERIVATIVE	AERO
DCAQ	R D	DERIVATIVE OF AXIAL FORCE WITH PITCH RATE	AERO
DCAQS	R D	SAVED VALUES OF AXIAL FORCE DERIVATIVE WITH PITCH RATE	AERO
DCA1	R U	EXPANSION ITERATION PARAMETER	EXPAND
DCA2	R U	EXPANSION ITERATION PARAMETER	EXPAND
DCLA	R D	DERIVATIVE OF LIFT COEFFICIENT WITH ANGLE OF ATTACK	AERO
DCLAS	R D	SAVED LIFT COEFFICIENT DERIVATIVE WITH ANGLE OF ATTACK	AERO
DCLD	R D	DERIVATIVE OF CL WITH CONTROL SURFACE DEFLECTION	AERO
DCLDS	R D	SAVED VALUES OF CL DERIVATIVE WITH CONTROL DEFLECTION	AERO
DCLLB	R D	DERIVATIVE OF ROLLING MOMENT WITH YAW ANGLE	AERO
DCLLBS	R D	SAVED VALUES OF ROLLING MOMENT DERIVATIVE WITH YAW	AERO
DCLLD	R D	DERIVATIVE OF ROLLING MOMENT WITH CONTROL DEFLECTION	AERO
DCLLDS	R D	SAVED VALUES OF ROLLING MOMENT-CONTROL DERIVATIVE	AERO
DCLLR	R D	DERIVATIVE OF ROLLING MOMENT WITH YAW RATE	AERO
DCLLRS	R D	SAVED VALUES OF ROLLING MOMENT DERIVATIVE WITH YAW	AERO
DCLND	R D	DERIVATIVE OF YAWING MOMENT WITH CONTROL DEFLECTION	AERO
DCLNDS	R D	SAVED VALUES OF YAWING MOMENT-CONTROL DERIVATIVE	AERO
DCLNR	R D	DERIVATIVE OF YAWING MOMENT WITH YAW RATE	AERO
DCLNRS	R D	SAVED VALUES OF YAWING MOMENT-YAW RATE DERIVATIVE	AERO
DCMA	R D	DERIVATIVE OF PITCHING MOMENT WITH ANGLE OF ATTACK	AERO
DCMADS	R D	SAVED VALUES OF PITCHING MOMENT-ALPHA DOT DERIVATIVE	AERO

SYMBOL	TYPE		DEFINITION	ROUTINE
DCMADT	R	D	PITCHING MOMENT-ALPHA DOT DERIVATIVE	AERO
DCMAS	R	D	SAVED VALUE OF PITCHING MOMENT-ALPHA DERIVATIVE	AERO
DCMD	R	D	PITCHING MOMENT-CONTROL DEFLECTION DERIVATIVE	AERO
DCMDS	R	D	SAVED VALUES OF PITCHING MOMENT-CONTROL DERIVATIVE	AERO
DCMQ	R	D	DERIVATIVE OF PITCHING MOMENT WITH PITCH RATE	AERO
DCMQS	R	D	SAVED VALUES OF PITCHING MOMENT-PITCH RATE DERIVATIVE	AERO
DCNA	R	D	DERIVATIVE OF NORMAL FORCE WITH ANGLE OF ATTACK	AERO
DCNAS	R	D	SAVED VALUE OF NORMAL FORCE-ALPHA DERIVATIVE	AERO
DCNB	R	D	DERIVATIVE OF YAWING MOMENT WITH YAW ANGLE	AERO
DCNBS	R	D	SAVED VALUE OF NORMAL FORCE-YAW DERIVATIVE	AERO
DCND	R	D	DERIVATIVE OF NORMAL FORCE WITH CONTROL DEFLECTION	AERO
DCNDS	R	D	SAVED VALUE OF NORMAL FORCE-CONTROL DERIVATIVE	AERO
DCNQ	R	D	DERIVATIVE OF NORMAL FORCE WITH PITCH RATE	AERO
DCNQS	R	D	SAVED VALUES OF NORMAL FORCE-PITCH RATE DERIVATIVES	AERO
DCYB	R	D	DERIVATIVE OF SIDE FORCE WITH YAW ANGLE	AERO
DCYBDS	R	D	SAVED VALUE OF CY-BETA DATA DERIVATIVE	AERO
DCYBDT	R	D	DERIVATIVE OF SIDE FORCE WITH BETA DOT	AERO
DCYBS	R	D	SAVED VALUE OF SIDE FORCE-YAW DERIVATIVE	AERO
DCYD	R	D	DERIVATIVE OF SIDE FORCE WITH CONTROL DEFLECTION	AERO
DCYDS	R	D	SAVED VALUES OF SIDE FORCE-CONTROL DERIVATIVE	AERO
DCYR	R	D	DERIVATIVE OF SIDE FORCE WITH YAW RATE	AERO
DCYRS	R	D	SAVED VALUES OF SIDE FORCE-YAW RATE DERIVATIVES	AERO
DD	R	D	TENTH DATA ARRAY	PLOT
DELCA	R	U	ELEMENT CONTRIBUTION TO AXIAL FORCE	FORCE
DELCA	R	U	DRAG COEFFICIENT INCREMENT	VECTOR
DELCLL	R	U	ELEMENT CONTRIBUTION TO ROLLING MOMENT	FORCE
DELCLL	R	U	ROLLING MOMENT COEFFICIENT INCREMENT	VECTOR
DELCLM	R	U	ELEMENT CONTRIBUTION TO PITCHING MOMENT	FORCE
DELCLM	R	U	PITCHING MOMENT COEFFICIENT INCREMENT	VECTOR
DELCLN	R	U	ELEMENT CONTRIBUTION TO YAWING MOMENT	FORCE
DELCLN	R	U	YAWING MOMENT COEFFICIENT INCREMENT	VECTOR
DELCN	R	U	ELEMENT CONTRIBUTION TO NORMAL FORCE	FORCE
DELCN	R	U	NORMAL FORCE COEFFICIENT INCREMENT	VECTOR
DELCPC	R	A	PRESSURE COEFFICIENT INCREMENT DUE TO CONTROL SURFACE	FLOSEP
DELCPC	R	U	DELTA CP DUE TO CONTROL SURFACE DEFLECTION	FORCE
DELCY	R	U	ELEMENT CONTRIBUTION TO SIDE FORCE	FORCE

SYMBOL	TYPE		DEFINITION	ROUTINE
DELCY	R	U	SIDE FORCE COEFFICIENT INCREMENT	VECTOR
DELDLW	R	U	IMPACT ANGLE FOR DELTA WING	FORCE
DELDLW	R	U	IMPACT ANGLE FOR DELTA-WING OPTION, RADIANS	SHKEXP
DELP	R	U	PRESSURE INCREMENT DUE TO FLAP AND SEPARATION	FLOSEP
DELQRP	R	U	INCREMENT IN ROTATION RATE FOR ROTATION DERIVATIVES	AERO
DELTA	R	U	EPSILON 1 + EPSILON 2	ANALY2
DELTA	R	A	ELEMENT IMPACT ANGLE (DEGREES)	FLOSEP
DELTA	R	U	IMPACT ANGLE, DEGREES	FORCE
DELTA	R	A	IMPACT ANGLE, DEGREES	SHKEXP
DELTA	R	A	IMPACT ANGLE, DEGREES	SKINFR
DELTAE	R	U	CONTROL SURFACE DEFLECTION	AERO
DELTAE	R	A	CONTROL SURFACE DEFLECTION	CONTRL
DELTAE	R	A	CONTROL SURFACE DEFLECTION	FLOSEP
DELTAE	R	A	CONTROL SURFACE DEFLECTION	FORCE
DELTAR	R	U	SURFACE IMPACT ANGLE IN RADIANS	CONE
DELTAR	R	A	ELEMENT IMPACT ANGLE (RADIANS)	FLOSEP
DELTAR	R	U	IMPACT ANGLE, RADIANS	FORCE
DELTAR	R	A	IMPACT ANGLE, RADIANS	SHKEXP
DELTAS	R	U	SAVED VALUE OF CONTROL SURFACE DEFLECTION	AERO
DELTAS	R	U	SAVED INITIAL CONTROL SURFACE DEFLECTION	FLOSEP
DELTAS	R	U	IMPACT CONTROL SURFACE DEFLECTION ANGLE	FORCE
DELTER	R	U	FLAP DEFLECTION ANGLE (RADIANS)	FLOSEP
DELTES	R	D	SAVED VALUE OF CONTROL SURFACE DEFLECTION	AERO
DELTH	R	U	THETA ANGULAR INCREMENT	ANALY1
DELTHB	R	U	BOTTOM SURFACE DELTA-THETA INCREMENT	SLABD
DELTHM	R	U	HINGE MOMENT INCREMENT	FORCE
DELTHT	R	U	TOP SURFACE DELTA THETA INCREMENT	SLABD
DELTHX	R	D	THETA ANGULAR INCREMENT ARRAY	ANALY1
DELTQ	R	U	IMPACT ANGLE AT MATCHING POINT	NEWTPM
DELTS	R	U	SAVED VALUES OF CONTROL DEFLECTION	AERO
DELTX	R	U	DISTANCE FROM LEADING EDGE TO CENTROID	SDATA
DELT2	R	U	ITERATION VALUE FOR EQUIVALENT CONE ANGLE (2)	FORCE
DELU	R	U	DELTA INCREMENT FOR U	ANALY2
DELVOL	R	U	ELEMENT VOLUME CONTRIBUTION	PICTUR
DELVOL	R	U	ELEMENT VOLUME CONTRIBUTION	SDATA
DELW	R	U	DELTA INCREMENT FOR W	ANALY2

SYMBOL	TYPE		DEFINITION	ROUTINE
DELX	R	U	GEOMETRY DATA X-INCREMENT	PICTUR
DELX	R	U	GEOMETRY DATA X-INCREMENT	SDATA
DELY	R	U	Y-SHIFT INCREMENT FOR ELLIPSE DATA	ANALY1
DELY	R	U	GEOMETRY DATA Y-INCREMENT	PICTUR
DELY	R	U	GEOMETRY DATA Y-INCREMENT	SDATA
DELYX	R	D	Y-SHIFT INCREMENT ARRAY	ANALY1
DELZ	R	U	Z-SHIFT INCREMENT FOR ELLIPSE DATA	ANALY1
DELZ	R	U	GEOMETRY DATA Z-INCREMENT	PICTUR
DELZ	R	U	GEOMETRY DATA Z-INCREMENT	SDATA
DELZ	R	U	GEOMETRY DATA DISPLACEMENT PARAMETER IN Z-DIRECTION	SLABD
DELZX	R	D	Z-SHIFT INCREMENT ARRAY	ANALY1
DEL1	R	U	ITERATION VALUE FOR EQUIVALENT CONE ANGLE (1)	FORCE
DISCON	I	U	ANGULAR MODE CONTROL FLAG	ANALY1
DIST1	R	U	LEADING EDGE DISTANCE VALUE	SDATA
DLTMU	R	U	EXPANSION ANGLE FROM MATCHING MOMENT	NEWTPM
DMDZ	R	U	DERIVATIVE OF MOLECULAR WEIGHT OF AIR	ATMOS
DO	R	U	BOUNDARY LAYER THICKNESS	FLOSEP
DOX	R	U	BOUNDARY LAYER THICKNESS AT EXACT SEPARATION POINT	FLOSEP
DO1	R	U	BOUNDARY LAYER THICKNESS ON ELEMENT BEFORE SEPARATION POINT	FLOSEP
DQC	R	U	DIFFERENCE IN CONVECTIVE HEATING RATES	TEMP
DQR	R	U	DIFFERENCE IN RADIATION HEATING RATES	TEMP
DS	R	U	SAVED VALUE OF SECOND INDEPENDENT VARIABLE	TTABLE
DSQXP	R	U	SQUARE ROOT OF X-PRIME	FLOSEP
DTC	R	U	DIFFERENCE IN CONVECTIVE TEMPERATURES	TEMP
DTR	R	U	DIFFERENCE IN RADIATION TEMPERATURES	TEMP
DUMMY	R	U	DUMMY ARGUMENT	FLOSEP
DUMMY	R	U	DUMMY VARIABLE	PLUNGE
DUMMY	R	U	DUMMY VARIABLE	SLABD
DX	R	U	INCREMENT BETWEEN VERTICAL GRID LINES	PLOT
DXEV	R	U	INCREMENT FROM ORIGIN, EVEN EXPONENTIAL	BLUNT
DXG	R	U	GRID DELTA-X INCREMENT	PICTUR
DXSEP	R	U	DIFFERENCE BETWEEN LEADING EDGE AND SEPARATION X-DISTANCE	FLOSEP
DXSEP1	R	U	XLE-XSEP ON ELEMENT JUST BEFORE SEPARATION ELEMENT	FLOSEP
DY	R	U	INCREMENT BETWEEN HORIZONAL GRID LINES	PLOT
DYG	R	U	GRID DELTA-Y INCREMENT	PICTUR
D1	R	U	UPSTREAM INTERACTION LENGTH	FLOSEP

SYMBOL	TYPE		DEFINITION	ROUTINE
D1DO	R	U	UPSTREAM INTERACTION LENGTH/BOUNDARY LAYER THICKNESS	FLOSEP
D2	R	U	DOWNSTREAM INTERACTION LENGTH TO PEAK PRESSURE	FLOSEP
D2D1	R	U	RATIO OF DOWNSTREAM TO UPSTREAM INTERACTION LENGTHS	FLOSEP
D3	R	U	DOWNSTREAM INTERACTION LENGTH TO PRESSURE RISE	FLOSEP
D3D1	R	U	DOWNSTREAM INTERACTION LENGTH/UPSTREAM INTERACTION LENGTH	FLOSEP
E	R	A	ELLIPTICAL INTEGRAL OF THE SECOND KIND	ELP1
E	R	D	FIFTH DATA ARRAY	PLOT
ECOS	R	U	COSINE OF FLOW TURNING ANGLE	SKINFR
EE	R	D	ELEVENTH DATA ARRAY	PLOT
EL	R	U	SURFACE REFERENCE LENTH, INPUT	SKINFR
EL	R	A	REFERENCE LENGTH	TEMP
ELAM	R	U	A FUNCTION OF RELATIVE FLAP CHORD LENGTH	FLOSEP
ELFI	R	U	FREE INTERACTION LENGTH	FLOSEP
ELFIDO	R	U	FREE INTERACTION LENGTH / BOUNDARY LAYER THICKNESS	FLOSEP
ELH	R	U	MOLECULAR SCALE TEMPERATURE DERIVATIVE, DEGREE RANKINE/FOOT	ATMOS
ELL	R	U	EFFECTIVE SURFACE LENGTH, LAMINAR	SKINFR
ELLOC	R	C	REFERENCE LENGTH	QC
ELLOC	R	C	REFERENCE LENGTH (=EL)	SKINFR
ELLOC	R	C	REFERENCE LENGTH (=EL)	TEMP
ELO	R	U	LENGTH OF INITIAL SURFACE	SKINFR
ELT	R	U	EFFECTIVE SURFACE LENGTH, TURBULENT	SKINFR
ELZ	R	U	MOLECULAR SCALE TEMPERATURE DERIVATIVE, DEGREE RANKINE/FOOT	ATMOS
EL1	R	U	EFFECTIVE LENGTH OF INITIAL SURFACE	SKINFR
EM	R	U	MOLECULAR WEIGHT OF AIR	ATMOS
EMADF	R	U	PRODUCT OF MACH NUMBER AND FLAP DEFLECTION	FLOSEP
EMCONE	R	U	MACH NUMBER ON SURFACE OF EQUIVALENT CONE	FORCE
EMISS	R	U	EMISSIVITY	TEMP
EMN	R	U	MACH NUMBER TIMES SINE THETA SQUARED	COMPR
EMN	R	U	MACH NUMBER NORMAL TO THE SHOCK	DELWNG
EMN	R	U	MACH NUMBER TIMES SHOCK ANGLE SQUARED	FORCE
EMN	R	A	MACH NUMBER TIMES SHOCK ANGLE SQUARED	NEWTPM
EMN	R	U	MACH NUMBER TIMES SHOCK ANGLE SQUARED	SKINFR
EMNS	R	U	MACH NUMBER NORMAL TO SHOCK	CONE
EMNS	R	U	MACH NUMBER NORMAL TO SHOCK	FORCE
EMNS	R	U	MACH NORMAL TO THE SHOCK	SHKEXP
EMSQ	R	U	MACH NUMBER SQUARED	EXPAND

SYMBOL	TYPE	DEFINITION	ROUTINE
EN	R U	PARAMETER IN CHARACTERISTIC LENGTH EQUATION	SKINFR
ENPM	R U	SURFACE SLOPE MODIFICATION FACTOR	AERO
ENPM	R A	SURFACE SLOPE MODIFICATION FACTOR	FORCE
ENPMS	R D	SAVED VALUES OF SURFACE SLOPE MODIFICATION FACTOR	AERO
ENTHAL	R U	ENTRY TO DETERMINE ENTHALPY	ROMU
EPS	R U	TOLERANCE FOR EVEN EXPONENTIAL	BLUNT
EPS	R U	ITERATION ACCURACY PARAMETER	EXPAND
EPSI	R U	EFFECTIVE DENSITY RATIO	NEWTPM
EPST	R U	TOLERANCE OF TEMPERATURE ITERATIONS	TEMP
EPS1	R U	EPSILON 1	ANALY2
EPS2	R U	EPSILON 2	ANALY2
ERFS	R U	ERROR FUNCTION PARAMETER	FORCE
ERROR	I C	ERROR FLAG	AERO
ERROR	I C	ERROR FLAG	ANALY1
ERROR	I C	ERROR FLAG	ANALY2
ERROR	I C	ERROR FLAG	COMPR
ERROR	I C	ERROR FLAG	CONE
ERROR	I C	ERROR FLAG	CONTRL
ERROR	I C	ERROR FLAG	DELWNG
ERROR	I C	ERROR FLAG	EXPAND
ERROR	I C	ERROR FLAG	FLOSEP
ERROR	I C	ERROR FLAG	FORCE
ERROR	I C	ERROR FLAG	GRAPIC
ERROR	I C	ERROR FLAG (=0 NO ERROR, =1 NON-FATAL, =2 FATAL)	MAIN
ERROR	I C	ERROR FLAG	NEWTPM
ERROR	I C	ERROR FLAG	PICTUR
ERROR	I C	ERROR FLAG	PLOT
ERROR	I C	ERROR FLAG	PLUNGE
ERROR	I C	ERROR FLAG	QC
ERROR	I C	ERROR FLAG	SDATA
ERROR	I C	ERROR FLAG	SHKEXP
ERROR	I C	ERROR FLAG	SKINFR
ERROR	I C	ERROR FLAG	SLABD
ERROR	I C	ERROR FLAG	TEMP
ERROR	I C	ERROR FLAG	VECTOR

SYMBOL	TYPE		DEFINITION	ROUTINE
ESIN	R	U	SINE OF FLOW TURNING ANGLE	SKINFR
ETA	R	D	COORDINATE IN ELEMENT COORDINATE SYSTEM	PICTUR
ETA	R	D	COORDINATE IN ELEMENT COORDINATE SYSTEM	SDATA
ETAC	R	U	PRANDTL-MEYER-EXPANSION CORRECTION FACTOR	AERO
ETAC	R	U	PRANDTL-MEYER EXPANSION CORRECTION FACTOR	COMPR
ETAC	R	A	PRANDTL-MEYER EXPANSION CORRECTION FACTOR	FORCE
ETAC	R	A	PRANDTL-MEYER EXPANSION CORRECTION FACTOR	NEW1PM
ETACK	R	U	ETA CHECK PARAMETER	PICTUR
ETACK	R	U	ETA CHECK PARAMETER	SDATA
ETACS	R	D	SAVED VALUES OF PRANDTL-MEYER CORRECTION FACTOR	AERO
ETACS	R	C	SAVED VALUES OF PRANDTL-MEYER CORRECTION FACTOR	SKINFR
ETACS	R	C	SAVED VALUES OF PRANDTL-MEYER CORRECTION FACTOR	TEMP
ETAO	R	U	COORDINATE IN ELEMENT COORDINATE SYSTEM	PICTUR
ETAO	R	U	CENTROID IN ELEMENT COORDINATE SYSTEM	SDATA
ETA2M4	R	U	CONSTANT IN AREA EQUATION	PICTUR
ETA2M4	R	U	CONSTANT IN AREA EQUATION	SDATA
EVK	R	U	EVEN EXPONENTIAL CONSTANT	BLUNT
EX	R	U	INDEPENDENT VARIABLE	BLUNT
F	R	D	SIXTH DATA ARRAY	PLOT
F	R	U	FORCE MAGNITUDE, POUNDS	VECTOR
FC	R	C	TURBULENT FLOW, SKIN FRICTION COMPRESSIBILITY FACTOR	QC
FC	R	C	TURBULENT FLOW, SKIN FRICTION COMPRESSIBILITY FACTOR	SKINFR
FC	R	C	TURBULENT FLOW, SKIN-FRICTION COMPRESSIBILITY FACTOR	TEMP
FF	R	D	TWELVETH DATA ARRAY	PLOT
FF	R	U	FRICTION FACTOR	SKINFR
FH	R	C	ENTHALPY ARRAY	DATA
FH	R	C	ENTHALPY ARRAY	KOMU
FH1	R	D	FIRST 108 ELEMENTS OF ENTHALPY ARRAY	DATA
FH2	R	D	FINAL 27 ELEMENTS OF ENTHALPY ARRAY	DATA
FJ	R	U	MANGLER TRANSFORMATION PARAMETER	SKINFR
FN	R	U	NORMAL MOMENTUM ACCOMODATION COEFFICIENT	FORCE
FOU	R	U	BLENDING FUNCTION VALUE	ANALY2
FOW	R	U	BLENDING FUNCTION VALUE	ANALY2
FR	R	C	DENSITY-VISCOSITY PRODUCT AND DENSITY ARRAYS	DATA
FR	R	C	DENSITY-VISCOSITY PRODUCT AND DENSITY ARRAYS	ROMU
FRX	R	C	TURBULENT FLOW, REYNOLDS NUMBER COMPRESSIBILITY FACTOR	QC

SYMBOL	TYPE		DEFINITION	ROUTINE
FRX	R	C	TURBULENT FLOW, REYNOLDS NUMBER COMPRESSIBILITY FACTOR	SKINFR
FRX	R	C	TURBULENT FLOW, REYNOLDS NUMBER COMPRESSIBILITY FACTOR	TEMP
FR1	R	D	DENSITY-VISCOSITY PRODUCT ARRAY	DATA
FR2	R	D	DENSITY ARRAY	DATA
FS	R	C	FLOW PROPERTIES BEFORE SHOCK OR EXPANSION	AERO
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	COMPR
FS	R	C	FLOW CONDITIONS BEFORE COMPRESSION	CONE
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	DELWNG
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	EXPAND
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	FLOSEP
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	FORCE
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	NEWTPM
FS	R	C	FREE-STREAM FLOW CONDITION ARRAY	QC
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	SHKEXP
FS	R	C	FLOW CONDITIONS BEFORE SHOCK OR EXPANSION	SKINFR
FS	R	C	FLOW CONDITIONS BEFORE THE SHOCK OR EXPANSION	TEMP
FS2	R	U	PRESSURE TO BE SAVED	FLOSEP
FS6	R	U	MACH NUMBER TO BE SAVED	FLOSEP
FT	R	U	TANGENTIAL MOMENTUM ACCOMODATION COEFFICIENT	FORCE
F1	R	U	FIRST FUNCTION OF EVEN EXPONENTIAL	BLUNT
F1U	R	U	BLENDING FUNCTION VALUE	ANALY2
F1W	R	U	BLENDING FUNCTION VALUE	ANALY2
G	R	U	GRAVITATIONAL ACCELERATION, FEET PER SECOND SQUARED	ATMOS
G	R	U	RATIO OF SPECIFIC HEATS	BLUNT
G	R	U	RATIO OF SPECIFIC HEATS =1.4	CONE
G	R	A	SPECIFIC HEAT RATIO (GAMMA)	FLOSEP
G	R	U	RATIO OF SPECIFIC HEATS = 1.4	FORCE
G	R	U	RATIO OF SPECIFIC HEATS = 1.4	NEWTPM
G	R	A	RATIO OF SPECIFIC HEATS	SHKEXP
G	R	U	RATIO OF SPECIFIC HEATS	SKINFR
G	R	U	RATIO OF SPECIFIC HEATS	TEMP
G	R	A	FIRST DERIVATIVE (NOT USED)	TTABLE
GAMMAT	R	U	TAIL DIHEDRAL ANGLE (DEGREES)	PLUNGE
GCP	R	U	GAS SPECIFIC HEAT AT CONSTANT PRESSURE	BLUNT
GCP	R	C	GAS SPECIFIC HEAT AT CONSTANT PRESSURE	QC
GCP	R	C	GAS SPECIFIC HEAT AT CONSTANT PRESSURE	SKINFR

SYMBOL	TYPE		DEFINITION	ROUTINE
GCP	R	C	GAS SPECIFIC HEAT AT CONSTANT PRESSURE	TEMP
GMRS	R	U	COMBINATION OF GEODETIC AND GAS CONSTANTS, DEG RANKINE/FOOT	ATMOS
GM1	R	U	RATIO OF SPECIFIC HEATS MINUS ONE	BLUNT
GP1	R	U	RATIO OF SPECIFIC HEATS PLUS ONE	BLUNT
GR	R	U	GAMMA RATIO FUNCTION	EXPAND
G0	R	U	GRAVITATIONAL ACCELERATION AT SEA LEVEL, FT/SEC SQUARED	ATMOS
G1	R	U	ACTUAL LOCATION ALONG X-AXIS FIRST PLOTTED POINT	PLOT
G1	R	A	SECOND DERIVATIVE (NOT USED)	TTABLE
G2	R	U	ACTUAL LOCATION ALONG X-AXIS OF SECOND PLOTTED POINT	PLOT
H	R	U	GEOPOTENTIAL ALTITUDE, FEET	ATMOS
HANKEY	R	U	NEWTONIAN CORRELATION FACTOR IN HANKEY EQUATION	FORCE
HAW	R	C	ADIABATIC-WALL ENTHALPY	QC
HAW	R	C	ADIABATIC-WALL ENTHALPY	SKINFR
HAW	R	C	ADIABATIC-WALL ENTHALPY	TEMP
HAWH1	R	U	ADIABATIC-WALL TO FREE-STREAM ENTHALPY RATIO	TEMP
HC1	R	U	FIRST ENTHALPY COEFFICIENT	ROMU
HC2	R	U	SECOND ENTHALPY COEFFICIENT	ROMU
HC3	R	U	THIRD ENTHALPY COEFFICIENT	ROMU
HG	R	D	MATRIX OF GEOPOTENTIAL ALTITUDES, FT	ATMOS
HLABEL	R	D	HORIZONTAL LABEL	PICTUR
HMFCT	R	U	HINGE MOMENT FACTOR	CONTRL
HMFCT	R	U	HINGE MOMENT FACTOR	FORCE
HML	R	D	HINGE MOMENT (+Y SIDE OF VEHICLE)	AERO
HML	R	A	HINGE MOMENT (+Y)	FORCE
HMLS	R	D	SAVED VALUES OF HINGE MOMENT (+Y)	AERO
HMLT	R	U	HINGE MOMENT (+Y)	AERO
HMR	R	D	HINGE MOMENT (-Y)	AERO
HMR	R	A	HINGE MOMENT (-Y)	FORCE
HMRS	R	D	SAVED VALUES OF HINGE MOMENT (-Y)	AERO
HMRT	R	U	HINGE MOMENT (-Y)	AERO
HS	R	A	ENTHALPY (FT/SEC)**2	ROMU
HSIMP	R	U	HYPERSONIC INTERACTION PARAMETER	FORCE
HSTAR	R	U	REFERENCE ENTHALPY	QC
HTITLE	R	D	VERTICAL LABEL	PICTUR

SYMBOL	TYPE	DEFINITION	ROUTINE
HTOT	R U	TOTAL ENTHALPY	TEMP
HW	R C	WALL ENTHALPY	QC
HW	R C	WALL ENTHALPY	SKINFR
HW	R C	WALL ENTHALPY	TEMP
HX	R A	INDEPENDENT VARIABLE	POLY
H1	R U	ACTUAL LOCATION ALONG Y-AXIS OF FIRST PLOTTED POINT	PLOT
H1	R C	FREE-STREAM ENTHALPY	QC
H1	R U	REDUCED ENTHALPY (HS*1.0E-8)	RONU
H1	R C	FREE-STREAM ENTHALPY	SKINFR
H1	R C	FREE-STREAM ENTHALPY	TEMP
H2	R U	ACTUAL LOCATION ALONG Y-AXIS SECOND PLOTTED POINT	PLOT
H2	R C	LOCAL ENTHALPY	QC
H2	R C	LOCAL ENTHALPY	SKINFR
H2	R C	LOCAL ENTHALPY	TEMP
I	I U	DO-LOOP INDEX	AERO
I	I U	SECTION INDEX NUMBER	ANALY1
I	I U	DO-LOOP INDEX	ANALY2
I	I U	DO LOOP INDEX WHEN DETERMINING APPROPRIATE ATMOSPHERE LAYER	ATMOS
I	I U	INDEX	CONTRL
I	I U	INDEX	ELP1
I	I U	ITERATION COUNTER	EXPAND
I	I U	DO LOOP COUNTER	FLOSEP
I	I U	DO-LOOP INDEX	MAIN
I	I U	ELEMENT NUMBER IN COLUMN	PICTUR
I	I D	PROGRAM CONTROL ARRAY	PLOT
I	I U	INDEX	PLUNGE
I	I A	INDEX NUMBER OF INITIAL COEFFICIENT	POLY
I	I U	ELEMENT NUMBER IN COLUMN	SDATA
I	I U	DO-LOOP INDEX (SKIN FRICTION SURFACE NUMBER)	SKINFR
I	I U	INDEX COUNTER	SLABD
I	I U	DO-LOOP INDEX	TTABLE
IA	I U	FLAG TO CONTROL SHIFTING OF COLUMN DATA POINTS FOR IORIEN=3	PICTUR
IA	I U	FLAG TO CONTROL SHIFTING OF COLUMN DATA POINTS FOR IORIEN=3	SDATA
IA	I U	INDEX	TTABLE
IABDOT	I U	ALPHA-DOT BETA-DOT DERIVATIVE FLAG	AERO
IAFLAG	I U	INPUT DATA READ CONTROL FLAG	ANALY2

SYMBOL	TYPE		DEFINITION	ROUTINE
IAREA	I	U	SURFACE AREA PRINT FLAG	PICTUR
IA1	I	U	INDEX	TTABLE
IB	I	U	FLAG TO CONTROL SHIFTING OF COLUMN DATA POINTS FOR IORIEN=2	PICTUR
IB	I	U	FLAG TO CONTROL SHIFTING OF COLUMN DATA POINTS FOR IORIEN=2	SDATA
IBFLAG	I	U	INPUT DATA READ CONTROL FLAG	ANALY2
IC	I	U	CAMERA SELECTION FLAG	PLOT
IC	I	U	INDEX	TTABLE
ICALC	I	U	CALCULATION OPTION FLAG	PLUNGE
ICHECK	I	U	PRINT FLAG FOR CHECKOUT PURPOSES	FLOSEP
ICNT	I	U	PICTURE COUNTER	PLOT
ICS	I	U	POINT CORRECT FLAG	PICTUR
IC1	I	U	INDEX	TTABLE
ID	I	D	INFORMATION ON TYPE 3 CARD FROM CC 73 TO CC 80	CARD
ID	I	U	INDEX	TTABLE
IDERIV	I	U	DERIVATIVE OPTION FLAG	AERO
IDERIV	I	A	DERIVATIVE CONTROL FLAG	FORCE
IDERIV	I	A	DERIVATIVE OPTION FLAG	PLUNGE
IDERS	I	D	SAVED VALUES OF DERIVATIVE OPTION FLAG	AERO
IDFLGA	I	U	ALPHA DERIVATIVE PRINT FLAG	AERO
IDFLGB	I	U	BETA DERIVATIVE PRINT FLAG	AERO
IDFLGC	I	U	ROLL DERIVATIVE PRINT FLAG (NOT USED BY MARK II)	AERO
IDFLGD	I	U	CONTROL DERIVATIVE PRINT FLAG	AERO
IDFLGE	I	U	PITCH RATE DERIVATIVE PRINT FLAG	AERO
IDFLGF	I	U	YAW RATE DERIVATIVE PRINT FLAG	AERO
IDSTAT	I	U	DERIVATIVE CYCLE FLAG	AERO
IDUM	I	U	DUMMY VARIABLE	PICTUR
ID1	I	U	INDEX	TTABLE
ID2	I	U	INDEX	TTABLE
IELOV	I	U	ELEMENT CHARACTERISTIC OVERRIDE FLAG	SDATA
IERR	I	U	ERROR FLAG	PICTUR
IERR	I	U	NAME OF AN ERROR LOCATION.	PLOT
IFACT	I	U	SCALE FACTOR FLAG	PICTUR
IFACT	I	U	SCALE FACTOR FLAG	SDATA
IFADV	I	U	FRAME FLAG	PICTUR
IFALSE	I	U	INPUT DATA READ CONTROL FLAG	ANALY2
IFIRST	I	U	FIRST POINT FLAG FOR USE IN NEWTPM	AERO
IFIRST	I	A	FIRST POINT FLAG FOR NEWTPM	COMPR

SYMBOL	TYPE		DEFINITION	ROUTINE
IFIRST	I	A	FLAG FOR FIRST TIME INTO NEWTONIAN-PRANDTL-MEYER ROUTINE	FLOSEP
IFIRST	I	A	INITIAL PRINT FLAG FOR NEWTPM	FORCE
IFIRST	I	A	FIRST POINT FLAG FOR USE IN NEWTPM	NEWTPM
IFIRST	I	A	FIRST POINT FLAG FOR USE IN NEWTPM	SHKEXP
IFIRST	I	A	INITIAL POINT FLAG FOR NEWTPM	SKINFR
IFLAG	I	U	CYCLE FLAG	CONTRL
IFLAG1	I	U	INPUT DATA READ CONTROL FLAG	ANALY2
IFLG	I	U	DERIVATIVE FLAG	AERO
IFLOW	I	C	LAMINAR (=1) OR TURBULENT (=2) FLOW FLAG	QC
IFLOW	I	C	LAMINAR (=1) OR TURBULENT (=2) FLOW FLAG	TEMP
IFRAME	I	U	FRAME ADVANCE FLAG	PICTUR
IFSCY	I	A	FLOW SEPARATION CYCLE FLAG	FLOSEP
IFSCY	I	U	CONTROL SURFACE FLOW SEPARATION CALCULATION CYCLE NUMBER	FORCE
IG	I	U	VERTICAL GRID LINE LABEL CONTROL FLAG	PICTUR
IGEOM	I	U	GEOMETRY SOURCE FLAG	SDATA
IGT	I	A	CONTROL SURFACE FLAG (=1 FORESURFACE, = 2 CONTROL SURFACE)	FLOSEP
IGT	I	U	CONTROL SURFACE FLAG (=1 FORE-SURFACE, =2 CONTROL SURFACE)	FORCE
IGT	I	U	CONTROL SURFACE FLAG	SDATA
IGTS	I	A	CONTROL SURFACE FLAG FOR THE PRESENT ELEMENT	FLOSEP
IGTS	I	U	CONTROL SURFACE FLAG FOR PRESENT ELEMENT	FORCE
IGTS	I	A	CONTROL SURFACE FLAG FOR PRESENT ELEMENT	SHKEXP
IGTY	I	U	INDUCED PRESSURE FLAG	SKINFR
IGTYPE	I	U	COMPONENT TYPE (=1 FOR CONTROL)	AERO
IGTYPE	I	A	GEOMETRY TYPE (=1 FOR CONTROL SURFACE COMPONENT)	FLOSEP
IGTYPE	I	A	GEOMETRY TYPE (=1 FOR CONTROL SURFACE COMPONENT)	FORCE
IGTYPE	I	A	COMPONENT TYPE FLAG	SDATA
IGTYPE	I	A	GEOMETRY TYPE (=1 FOR CONTROL SURFACE COMPONENT)	SHKEXP
IGTYPE	I	A	GEOMETRY TYPE	SKINFR
IH	I	U	TERM CAUSES LABEL OF EVERY IHTH HORIZONTAL GRID LINE	PLOT
IH	I	O	ENTHALPY ARRAY INDEX	ROMU
IHM	I	U	ELEMENT NUMBER INDEX	CONTRL
IHM	I	U	HINGE MOMENT ELEMENT INDEX	FORCE
IHW	I	C	WALL ENTHALPY FLAG	QC
IHW	I	C	WALL ENTHALPY FLAG	TEMP
IH1	I	U	ENTHALPY ARRAY INDEX AT FIRST PRESSURE	ROMU
IH2	I	U	ENTHALPY ARRAY INDEX AT SECOND PRESSURE	ROMU

SYMBOL	TYPE		DEFINITION	ROUTINE
IH3	I	U	ENTHALPY ARRAY INDEX AT THIRD PRESSURE	ROHU
II	I	U	BOUNDARY NUMBER	ANALY2
II	I	U	HORIZONTAL TITLE ARRAY SUBSCRIPT	PLOT
II	I	U	NUMBER OF ELEMENTS IN COLUMN	SDATA
II	I	U	LEADING ELEMENT INDEX	SHKEXP
II	I	U	DO-LOOP INDEX (SKIN FRICTION SURFACE NUMBER)	SKINFR
IIA	I	U	DATA SHIFTING CONTROL PARAMETER (IORIEN=3)	PICTUR
IIA	I	U	DATA SHIFTING CONTROL PARAMETER (IORIEN=3)	SDATA
IIB	I	U	DATA SHIFTING CONTROL PARAMETER (IORIEN=2)	PICTUR
IIB	I	U	DATA SHIFTING CONTROL PARAMETER (IORIEN=2)	SDATA
IL	I	U	NUMBER OF ELEMENTS ON THE FORE SURFACE	CONTRL
IM	I	C	ELEMENT ROW NUMBER ARRAY	AERO
IM	I	C	ELEMENT ROW NUMBER ARRAY	COMPR
IM	I	C	ELEMENT ROW NUMBER ARRAY	CONE
IM	I	C	ELEMENT ROW NUMBER ARRAY	CONTRL
IM	I	C	ELEMENT ROW NUMBER ARRAY	DELWNG
IM	I	C	ELEMENT ROW NUMBER ARRAY	EXPAND
IM	I	C	(NOT USED)	FLOSEP
IM	I	C	ELEMENT ROW NUMBER ARRAY	FORCE
IM	I	C	ELEMENT ROW NUMBER ARRAY	NEWTPM
IM	I	C	ELEMENT ROW NUMBER ARRAY	QC
IM	I	C	ELEMENT ROW NUMBER ARRAY	SDATA
IM	I	C	ELEMENT ROW NUMBER ARRAY	SHKEXP
IM	I	C	ELEMENT ROW NUMBER ARRAY	SKINFR
IM	I	C	ELEMENT ROW NUMBER ARRAY	TEMP
IMP	I	D	IMPACT METHOD ARRAY	AERO
IMPACI	I	U	STARTING ELEMENT IMPACT METHOD	AERO
IMPACI	I	A	INITIAL STRIP ELEMENT IMPACT FORCE METHOD	FLOSEP
IMPACI	I	A	STARTING ELEMENT IMPACT METHOD	FORCE
IMPACI	I	A	STARTING ELEMENT IMPACT METHOD	SHKEXP
IMPACT	I	U	IMPACT FORCE CALCULATION METHOD	AERO
IMPACT	I	A	IMPACT FORCE CALCULATION METHOD FLAG	FLOSEP
IMPACT	I	A	IMPACT FORCE CALCULATION METHOD	FORCE
IMPI	I	D	STARTING IMPACT METHOD ARRAY	AERO
IMPS	I	A	INPUT IMPACT FORCE CALCULATION METHOD	FLOSEP
IMPS	I	U	SAVED VALUE OF IMPACT FORCE METHOD	FORCE

SYMBOL	TYPE		DEFINITION	ROUTINE
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	AERO
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	COMPR
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	CONE
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	CONTRL
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	DELWNG
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	EXPAND
IN	I	C	IN(1) AND IN(2) NUMBER OF ELEMENTS IN FORE-SURFACE AND FLAP	FLOSEP
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	FORCE
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	NEWTPM
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	QC
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	SDATA
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	SHKEXP
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	SKINFR
IN	I	C	ELEMENT COLUMN NUMBER ARRAY	TEMP
INU	I	U	FLAG	ANALY2
IORIEN	I	U	ELEMENT ORIENTATION	AERO
IORIEN	I	A	ELEMENT ORIENTATION (=1 FOR STREAMWISE STRIP)	FLOSEP
IORIEN	I	A	ELEMENT ORIENTATION	FORCE
IORIEN	I	U	ELEMENT ORIENTATION (NOT USED)	PICTUR
IORIEN	I	A	ELEMENT ORIENTATION FLAG	SDATA
IORIEN	I	A	ELEMENT ORIENTATION	SHKEXP
IP	I	U	NUMBER OF CHARACTERS TO BE USED AS PLOTTING SYMBOLS	PLOT
IP	I	U	CARD PRINT POSITION FLAG	SLABD
IPART	I	U	CONTROL FLAG FOR COMPONENT TYPE (BODY,WING,TAIL)	PLUNGE
IPG	I	O	PROGRAM OPTION FLAG ARRAY	MAIN
IPIC	I	U	FRAME NUMBER	PICTUR
IPL10	I	U	TYPE NUMBER FOR TAPE 10 DATA = 42	AERO
IPL9	I	U	TYPE NUMBER FOR TAPE 9 DATA =43	AERO
IPRCK	I	A	DETAILED DATA PRINT CONTROL FLAG	FLOSEP
IPRCK	I	U	PRINT FLAG	FORCE
IPRCK	I	A	PRINT FLAG	SHKEXP
IPRINS	I	O	SAVED VALUES OF PRINT FLAG	AERO
IPRINT	I	U	PRINT FLAG FOR SHOCK-EXPANSION CALCULATIONS	AERO
IPRINT	I	U	PRINT FLAG	ANALY1
IPRINT	I	U	PRINT FLAG	ANALY2
IPRINT	I	A	PRINT FLAG	COMPR

SYMBOL	TYPE		DEFINITION	ROUTINE
IPRINT	I	A	PRINT FLAG	EXPAND
IPRINT	I	A	PRINT FLAG	FORCE
IPRINT	I	A	PRINT FLAG	NEMTPM
IPRINT	I	A	PRINT FLAG	PUNCH
IPRINT	I	A	PRINT FLAG	SHKEXP
IPRINT	I	U	PRINT FLAG	SKINFR
IPRINT	I	U	PRINT FLAG	SLABD
IPRINT	I	A	PRINT FLAG	TEMP
IPRINT	I	U	PRINT FLAG	VECTOR
IPROG	I	A	PROGRAM OPTION NUMBER	GRAPIC
IPROG	I	U	ACTIVE PROGRAM OPTION	MAIN
IPRTS	I	D	SAVED VALUES OF IPRINT FLAG	AERO
IPS	I	U	SC-4020 AERO-DATA SAVE FLAG	AERO
IQUAD	I	U	QUADRILATERAL PLOT FLAG	PICTUR
IR	I	U	CONTROL FLAG VALUE OF WHICH DETERMINES NEXT OPERATION	PLOT
IRFD11	I	U	TAPE 11 READ FLAG INDICATOR	FORCE
IRFD11	I	A	TAPE 11 READ FLAG INDICATOR	FLOSEP
IREFL	I	U	REFLECTION ELEMENTS CONTROL FLAG	PICTUR
IRET1	I	U	RETURN TO TYPE 1 CARD CONTROL FLAG	AERO
IREW8	I	U	REWIND TAPE 8 FLAG	AERO
IREW8	I	U	TAPE 8 REWIND FLAG	PICTUR
IREW8	I	U	TAPE 8 REWIND FLAG	SLABD
IRFLAG	I	U	INPUT DATA READ CONTROL FLAG	ANALY2
IRFLG	I	U	REFLECTION CONTROL FLAG	PICTUR
IRM1	I	U	DENSITY-VISCOSITY ARRAY INDEX AT 10.0 ATM.	ROMU
IRM2	I	U	DENSITY-VISCOSITY ARRAY INDEX AT 10.0**-4 ATM.	ROMU
IRO1	I	U	DENSITY ARRAY INDEX AT 10.0 ATM.	ROMU
IRO2	I	U	DENSITY ARRAY INDEX AT 10.0**-4 ATM.	ROMU
IS	I	C	SKIN FRICTION CONTROL FLAG ARRAY	AERO
IS	I	C	SKIN FRICTION FLAG DATA ARRAY	FORCE
IS	I	U	PLOTTING SYMBOL CODE	PLOT
IS	I	C	SKIN FRICTION FLAG DATA ARRAY	SKINFR
IS	I	C	SKIN FRICTION FLAG DATA ARRAY	TEMP
ISBP	I	A	FORCE SUMMATION BYPASS FLAG (=1 TO BYPASS SUMMATION)	FLOSEP
ISBP	I	U	FORCE SUMMATION BY PASS FLAG (=1 TO BY PASS SUMMATION)	FORCE
ISCT	I	U	ELEMENT COUNTER	FLOSEP
ISDET	I	A	DATA GENERATION CONTROL FLAG	COMPR
ISDET	I	A	DATA GENERATION CONTROL FLAG	CONE

SYMBOL	TYPE		DEFINITION	ROUTINE
ISDET	I	A	DATA GENERATION CONTROL FLAG	EXPAND
ISDET	I	U	CALCULATION CONTROL FLAG FOR COMPRESSION ROUTINE	FLOSEP
ISDET	I	U	DATA GENERATION CONTROL FLAG	FORCE
ISDET	I	U	DATA GENERATION CONTROL FLAG	NEWTPM
ISDET	I	U	DATA GENERATION CONTROL FLAG	SHKEXP
ISDET	I	U	DATA GENERATION CONTROL FLAG	SKINFR
ISE	I	U	DATA GENERATION CONTROL FLAG	COMPR
ISE	I	U	DATA GENERATION CONTROL FLAG	FORCE
ISE	I	A	DATA GENERATION CONTROL FLAG	NEWTPM
ISE	I	U	DATA GENERATION CONTROL FLAG	SKINFR
ISEP	I	U	SEPARATION INDICATOR FLAG	FLOSEP
ISH	I	D	SHADOW METHOD ARRAY	AERO
ISHAD	I	U	SHADOW FORCE CALCULATION METHOD	AERO
ISHAD	I	A	SHADOW FORCE CALCULATION METHOD FLAG	FLOSEP
ISHAD	I	A	SHADOW FORCE CALCULATION METHOD	FORCE
ISHAD	I	U	SHADOW ELEMENT FLAG	PICTUR
ISHADI	I	U	STARTING ELEMENT METHOD IN SHADOW REGION	AERO
ISHADI	I	A	INITIAL STRIP ELEMENT SHADOW FORCE METHOD	FLOSEP
ISHADI	I	A	STARTING ELEMENT METHOD IN SHADOW REGION	FORCE
ISHADI	I	A	STARTING ELEMENT METHOD IN SHADOW REGION	SHKEXP
ISHI	I	D	SHADOW STARTING ELEMENT ARRAY	AERO
ISHS	I	A	INPUT SHADOW FORCE CALCULATION METHOD	FLOSEP
ISHS	I	U	SAVED VALUE OF SHADOW FORCE METHOD	FORCE
ISIDE	I	U	GEOMETRY ANGULAR POSITION INDICATOR	SLABO
ISIZ	I	U	NUMBER OF ELEMENTS TO BE STORED IN CORE	AERO
ISIZ	I	A	NUMBER OF ELEMENTS STORED IN CORE	CONTRL
ISIZ	I	A	NUMBER OF ELEMENTS TO BE STORED IN CORE	FORCE
ISIZ	I	A	NUMBER OF ELEMENTS TO BE STORED IN CORE	SDATA
ISIZ	I	A	NUMBER OF ELEMENTS STORED IN CORE	SKINFR
ISMODE	I	A	SHOCK-EXPANSION MODE FLAG (USED IN FLOSEP)	SHKEXP
ISOVR	I	U	FIRST POINT STATUS OVERRIDE FLAG	ANALY2
ISPNT	I	A	SEPARATION AND ATTACHMENT PRINT FLAG	FLOSEP
ISPNT	I	U	SEPARATION AND ATTACHMENT PRINT FLAG	FORCE
ISTART	I	U	CONTROL FLAG	PICTUR
ISTAT	I	U	SURFACE POINT STATUS FLAG	ANALY2
ISTATT	I	U	SURFACE POINT STATUS FLAG	ANALY2

SYMBOL	TYPE		DEFINITION	ROUTINE
ISTAT3	I	U	NUMBER OF STATUS = 3 POINTS IN DECK	PICTUR
ISUM	I	U	COMPONENT SUMMATION AND SAVE FLAG	AERO
IS3	I	U	PRESSURE CALCULATION METHOD FLAG	SKINFR
IT	I	U	TYPE OF INPUT DATA CARD	PLOT
ITAPE	I	U	GEOMETRY SOURCE FLAG	PICTUR
ITAPE	I	U	GEOMETRY TAPE CONTROL FLAG	SDATA
ITOC	I	U	THICKNESS CORRECTION TABLE FLAG	SLABD
ITOP	I	U	TOP GEOMETRY CONTROL FLAG	SLABD
ITRANS	I	U	FLOW TRANSITION FLAG (=0 LAMINAR, =1 TURBULENT)	FLOSEP
ITRUE	I	U	INPUT DATA CONTROL FLAG	ANALY2
ITURB	I	C	TURBULENT FLOW FLAG	QC
ITURB	I	C	TURBULENT FLOW FLAG	TEMP
ITW	I	C	IDEAL GAS (=1) OR REAL GAS (=2) FLAG	QC
ITW	I	C	IDEAL GAS (=1) OR REAL GAS(=2) FLAG	TEMP
ITYPE	I	U	CARD TYPE	ANALY1
ITYPE	I	U	CARD TYPE NUMBER	ANALY2
ITYPE	I	U	FLAG TO CONTROL EQUATION TO BE USED IN CALCULATING KBW	PLUNGE
ITYPE	I	U	CARD TYPE	SLABD
ITYP13	I	U	TYPE 13 CARD READ FLAG	AERO
IV	I	U	TERM CAUSES LABEL OF EVERY IVTH VERTICAL GRID LINE	PLOT
IVCTNO	I	U	VECTOR NUMBER	VECTOR
IVECT	I	U	FORCE VECTOR METHOD FLAG	AERO
IVISIN	I	A	VISCOUS-INTERACTION CONTROL FLAG	BLUNT
IVISIN	I	U	VISCOUS-INTERACTION CONTROL FLAG	FORCE
IX1	I	U	X-RASTER COORDINATE OF FIRST POINT	PICTUR
IX1	I	U	RASTER LOCATION ALONG X-AXIS OF FIRST PLOTTED POINT	PLOT
IX2	I	U	X-RASTER COORDINATE OF SECOND POINT	PICTUR
IX2	I	U	RASTER LOCATION ALONG X-AXIS OF SECOND PLOTTED POINT	PLOT
IY	I	U	VERTICAL RASTER LOCATION OF HORIZONTAL TITLES	PLOT
IY1	I	U	Y-RASTER COORDINATE OF FIRST POINT	PICTUR
IY1	I	U	RASTER LOCATION ALONG Y-AXIS SECOND PLOTTED POINT	PLOT
IY2	I	U	Y-RASTER COORDINATE OF SECOND POINT	PICTUR
IY2	I	U	RASTER LOCATION ALONG Y-AXIS OF SECOND PLOTTED POINT	PLOT
I4CT	I	U	COUNTER FOR READING CONTROL SURFACE GEOMETRY DATA	FORCE
I8BSP	I	U	TAPE 8 BACKSPACE CONTROL FLAG	SLABD
J	I	U	DO-LOOP INDEX	AERO

SYMBOL	TYPE		DEFINITION	ROUTINE
J	I	U	ANGULAR INCREMENT COUNTER	ANALY1
J	I	U	COUNTER IN VARIOUS DO LOOPS	ATMOS
J	I	U	DO-LOOP INDEX	CONTRL
J	I	U	DO-LOOP INDEX	ELP1
J	I	U	DO LOOP INDEX	FLOSEP
J	I	A	ALPHA-BETA COUNTER FLAG	FORCE
J	I	U	INDEX ON PROGRAM OPTION	MAIN
J	I	U	ITERATION COUNTER	NEWTPM
J	I	U	MULTI-PURPOSE INDEX	PLOT
J	I	U	DO-LOOP INDEX	POLY
J	I	U	ENTHALPY INDEX COUNTER	ROMU
J	I	U	ANGULAR LOOP INDEX	SLABD
J	I	U	INDEX	TTABLE
JG	I	U	HORIZONTAL GRID LINE LABEL CONTROL FLAG	PICTUR
JJ	I	U	COUNTER	SDATA
JPATH	I	U	CONTROL FLAG FOR ITERATION PATH	EXPAND
J1	I	U	POINT INDEX	ANALY2
J2	I	U	POINT INDEX	ANALY2
J2	I	U	INDEX	TTABLE
J3	I	U	POINT INDEX	ANALY2
K	I	U	CARD WRITE CYCLE FLAG	ANALY1
K	I	U	DO-LOOP INDEX	ANALY2
K	I	U	COUNTER IN DO LOOP DETERMINING APPROPRIATE ATMOSPHERE LAYER	ATMOS
K	I	C	NUMBER OF ELEMENTS	COMPR
K	I	C	NUMBER OF ELEMENTS	CONE
K	I	C	NUMBER OF ELEMENTS	DELWNG
K	I	C	NUMBER OF ELEMENTS	EXPAND
K	I	C	NUMBER OF ELEMENTS IN COMPONENT	FLOSEP
K	I	C	NUMBER OF ELEMENTS	FORCE
K	I	C	NUMBER OF ELEMENTS	NEWTPM
K	I	U	MULTI-PURPOSE INDEX	PLOT
K	I	U	FLAG TO CONTROL SELECTION OF KBW EQUATION	PLUNGE
K	I	U	COEFFICIENT NUMBER	POLY
K	I	C	NUMBER OF ELEMENTS	SHKEXP
K	I	U	FLAG (=1 LAMINAR, =2 TURBULENT)	TEMP
K	I	U	DO-LOOP INDEX	TTABLE

SYMBOL	TYPE		DEFINITION	ROUTINE
KBW	R	U	INTERFERENCE ON BODY IN PRESENCE OF WING/TAIL	PLUNGE
KF	I	U	METHOD-SURFACE TYPE FLAG	SKINFR
KFLSP	I	U	FLOW SEPARATION FLAG (=0 NO SEPARATION, =1 FLOW SEPARATED)	FLOSEP
KKXY	I	U	OFF-SCALE DETECTION FLAG	PICTUR
KLCT	I	U	COUNTER	PICTUR
KLCT	I	U	COUNTER	SDATA
KQ	R	U	SIMILARITY PARAMETER	SKINFR
KSUB	I	U	SECONDARY COUNTER IN TEMPERATURE ITERATIONS	TEMP
KT	I	U	TEMPERATURE ITERATION COUNTER	TEMP
KTMAX	I	U	MAXIMUM NUMBER OF TEMPERATURE ITERATIONS	TEMP
KWB	R	U	INTERFERENCE ON WING/TAIL IN PRESENCE OF BODY	PLUNGE
KX	I	U	OFF-SCALE DETECTION FLAG	PICTUR
KX1	I	U	OFF-SCALE DETECTION FLAG	PICTUR
KY	I	U	OFF-SCALE DETECTION FLAG	PICTUR
KY1	I	U	OFF-SCALE DETECTION FLAG	PICTUR
K2	I	U	INDEX	TTABLE
L	I	C	NUMBER OF ELEMENTS	AERO
L	I	U	INDEX	ANALY2
L	I	U	FLAG	COMPR
L	I	C	NUMBER OF ELEMENTS	CONTRL
L	I	A	NUMBER OF ELEMENT IN FORCE CALCULATION LOOP	FLOSEP
L	I	U	ELEMENT FORCE CALCULATION LOOP COUNTER	FORCE
L	I	U	ELEMENT NUMBER	PICTUR
L	I	U	MULTI-PURPOSE INDEX AND FILM ADVANCE FLAG	PLOT
L	I	C	NUMBER OF ELEMENTS	QC
L	I	C	NUMBER OF ELEMENTS	SDATA
L	I	C	NUMBER OF ELEMENTS	SKINFR
L	I	C	NUMBER OF ELEMENTS	TEMP
L	I	U	INDEX	TTABLE
LAMBDA	R	U	WING/TAIL TAPER RATIO (TIP CHORD/ROOT CHORD)	PLUNGE
LAMBDA	R	U	MODIFIED HYPERSONIC INTERACTION PARAMETER	SKINFR
LAST	I	U	LAST FLIGHT CONDITION FLAG	AERO
LAST	I	U	LAST CROSS-SECTION FLAG	ANALY1
LAST	I	U	LAST FLAG	ANALY2
LAST	I	U	LAST PLOT CONTROL FLAG	PICTUR
LAST	I	A	LAST FLAG	PUNCH

SYMBOL	TYPE		DEFINITION	ROUTINE
LAST	I	U	SLAB DELTA OPTION TERMINATION FLAG	SLABD
LAST	I	U	LAST VECTOR FLAG	VECTOR
LAST2	I	U	FLAG TO INDICATE LAST CARD OF T/C TABLE	SLABD
LAST3	I	U	FLAG TO INDICATE LAST CROSS-SECTION CARD	SLABD
LATT	I	U	ELEMENT NUMBER AT FLOW ATTACHMENT POINT	FLOSEP
LEFCT	R	U	LEADING EDGE FACTOR	SDATA
LENGTH	R	U	BODY LENGTH	PLUNGE
LIM	I	U	ANGULAR LIMIT FLAG	ANALY1
LINE	I	U	LINE COUNTER	ANALY1
LINE	I	U	LINE COUNTER	ANALY2
LINE	I	A	LINE COUNTER	PUNCH
LINE	I	U	OUTPUT LINE COUNTER	SLABD
LL	I	U	ELEMENT NUMBER	CONTRL
LL	I	A	ELEMENT NUMBER	FLOSEP
LL	I	U	ELEMENT NUMBER	FORCE
LL	I	A	ELEMENT NUMBER	SHKEXP
LN	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
LOD	R	A	LIFT-TO-DRAG RATIO	VECTOR
LOVERD	R	U	LIFT TO DRAG RATIO	FORCE
LS	I	C	NUMBER OF ELEMENTS	AERO
LS	I	C	NUMBER OF ELEMENTS	COMPR
LS	I	C	NUMBER OF ELEMENTS	CONE
LS	I	C	NUMBER OF ELEMENTS	CONTRL
LS	I	C	NUMBER OF ELEMENTS	DELWNG
LS	I	C	NUMBER OF ELEMENTS	EXPAND
LS	I	C	NUMBER OF ELEMENTS IN COMPONENT	FLOSEP
LS	I	C	NUMBER OF ELEMENTS	FORCE
LS	I	C	NUMBER OF ELEMENTS	NEWTPH
LS	I	C	NUMBER OF ELEMENTS	OC
LS	I	C	NUMBER OF ELEMENTS	SDATA
LS	I	C	NUMBER OF ELEMENTS	SHKEXP
LS	I	C	NUMBER OF ELEMENTS	SKINFR
LS	I	C	NUMBER OF ELEMENTS	TEMP
LSAVE	I	U	SAVEL ELEMENT NUMBER (ALSO ITERATION COUNTER)	FORCE
LSEP	I	U	ELEMENT NUMBER AT SEPARATION	FLOSEP
LSS	I	U	NUMBER OF ELEMENTS IN COMPONENT	FLOSEP

SYMBOL	TYPE		DEFINITION	ROUTINE
LXY	R	U	HINGE LINE LENGTH IN X-Y PLANE	CONTRL
LYZ	R	U	HINGE LINE LENGTH IN Y-Z PLANE	CONTRL
L1	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
L2	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
L21	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
L31	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
L32	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
M	I	U	COUNTER	ANALY1
M	I	U	DATA READ IN CONTROL FLAG	ANALY2
M	I	U	ELEMENT ROW NUMBER	CONTRL
M	I	A	ELEMENT NUMBER IN STRIP	FLOSEP
M	I	U	ELEMENT ROW NUMBER	FORCE
M	I	U	DATA READ IN CONTROL FLAG	PICTUR
M	I	U	MULTI-PURPOSE INDEX AND EMPHASIS FLAG FOR HORIZ. GRID LINES	PLOT
M	R	U	COTANGENT OF WING/TAIL LEADING EDGE SWEEP ANGLE	PLUNGE
M	I	U	DATA READ IN CONTROL FLAG	SDATA
M	I	A	ELEMENT ROW NUMBER	SHKEXP
M	R	U	SLOPE OF PRESSURE RATIO VERSUS LAMBDA	SKINFR
M	I	U	COUNTER	SLABD
MA	R	U	LOCAL MACH NUMBER ON ELEMENT	FLOSEP
MAC	R	U	REFERENCE LENGTH FOR MOMENT COEFFICIENTS	AERO
MAC	R	A	REFERENCE LENGTH FOR MOMENT COEFFICIENTS	FORCE
MAC	R	A	REFERENCE LENGTH FOR MOMENT COEFFICIENTS	VECTOR
MACH	R	U	MACH NUMBER	AERO
MACH	R	A	MACH NUMBER	BLUNT
MACH	R	U	MACH NUMBER	CONE
MACH	R	A	FREE-STREAM MACH NUMBER	FLOSEP
MACH	R	A	MACH NUMBER	FORCE
MACH	R	U	MACH NUMBER	NEWTPM
MACH	R	A	MACH NUMBER	SHKEXP
MACH	R	A	MACH NUMBER	SKINFR
MACH	R	A	MACH NUMBER	VECTOR
MACHH	R	U	SHOCK-EXPANSION MACH NUMBER AT HINGE LINE ELEMENT	FLOSEP
MACHI	R	D	STARTING OR PREVIOUS ELEMENT MACH NUMBER	SHKEXP
MACHO	R	U	MACH NUMBER	COMPR
MACHO	R	U	INITIAL MACH NUMBER	NEWTPM

SYMBOL	TYPE		DEFINITION	ROUTINE
MACHSQ	R	U	MACH NUMBER SQUARED	COMPR
MACHSQ	R	U	SQUARE OF MACH NUMBER NORMAL TO EFFECIVE SHOCK	NEWTPM
MACHX	R	U	MACH NUMBER AT SEPARATION	FLOSEP
MACHX1	R	U	MACH NUMBER ON ELEMENT JUST BEFORE SEPARATION	FLOSEP
MACH1	R	U	MACH NUMBER	COMPR
MARKPT	I	U	PLOTTING SYMBOL CODE	PICTUR
MC	I	U	DATA READ IN CONTROL NUMBER	ANALY2
MC	I	U	DATA READ IN CONTROL NUMBER	PICTUR
MC	I	U	DATA READ IN CONTROL NUMBER	SDATA
MER	I	A	ERROR FLAG	COMPR
MER	I	A	ERROR FLAG	EXPAND
MER	I	U	COMPRESSION ROUTINE ERROR FLAG	FLOSEP
MER	I	U	ERROR FLAG	FORCE
MER	I	A	ERROR FLAG	NEWTPM
MER	I	U	ERROR FLAG	SHKEXP
MER	I	U	ERROR FLAG	SKINFR
MER	I	A	ERROR FLAG	TEMP
MEREXP	I	U	VACUUM EXPANSION FLAG	SKINFR
MG	I	U	HORZONTAL LINE EMPHASIZE FLAG	PICTUR
ML	I	U	DATA READ IN CONTROL NUMBER	ANALY2
MMAX	I	U	MAXIMUM VALUE FOR PARAMETER M	SHKEXP
MMIN	I	U	NUMBER OF ELEMENTS IN A COLUMN	PICTUR
MMIN	I	U	NUMBER OF ELEMENTS IN A COLUMN	SDATA
MN	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
MODE	I	U	GEOMETRY MODE FLAG	SLABD
MSUBQ	R	U	MACH NUMBER AT MATCHING POINT	NEWTPM
MU	R	U	VISCOSITY	NEWTPM
M1	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
M1	R	U	FIRST ITERATION MACH NUMBER	NEWTPM
M1	I	U	INDEX	TTABLE
M2	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
M2	R	U	SECOND ITERATION MACH NUMBER	NEWTPM
M2	I	U	INDEX	TTABLE
N	I	U	COUNTER	ANALY1
N	I	U	DATA READ IN COUNTER	ANALY2
N	I	U	ELEMENT COLUMN NUMBER	CONTRL

SYMBOL	TYPE		DEFINITION	ROUTINE
N	I	A	STREAMWISE ELEMENT STRIP NUMBER	FLOSEP
N	I	U	ELEMENT COLUMN NUMBER	FORCE
N	I	U	COLUMN NUMBER	PICTUR
N	I	U	MULTI-PURPOSE INDEX AND EMPHASIS FLAG FOR VERT. GRID LINES	PLOT
N	I	A	ORDER OF POLYNOMIAL	POLY
N	I	U	ORDER OF POLYNOMIAL	ROMU
N	I	U	COLUMN NUMBER	SDATA
N	I	A	ELEMENT COLUMN NUMBER	SHKEXP
N	I	U	COUNTER	SLABD
NAB	I	U	NUMBER OF ALPHA-BETA COMBINATIONS	AERO
NABCT	I	U	ALPHA-BETA COUNTER	AERO
NABS	I	U	SAVED VALUE FOR NUMBER OF ALPHA-BETA COMBINATIONS	AERO
NB	I	U	BOUNDARY CURVE NUMBER	ANALY2
NC	I	U	NUMBER OF DATA POINTS	PLOT
NC	I	U	NOT USED	SKINFR
NCAM	I	U	CAMERA SELECTION FLAG	PICTUR
NERROR	I	A	ERROR FLAG	ELP1
NER1	I	U	ERROR FLAG	PLUNGE
NFS	I	U	ELEMENT COUNTER	SDATA
NG	I	U	VERTICAL LINE EMPHASIZE FLAG	PICTUR
NM	I	U	COUNTER	ANALY1
NN	I	D	COUNTER	ANALY1
NN	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
NN	R	U	SURFACE NORMAL LENGTH	CONTRL
NN	I	U	COLUMN ELEMENT COUNTER	PICTUR
NN	I	U	ELEMENT COUNTER	SDATA
NN2	I	U	COUNTER	PICTUR
NN2	I	U	ELEMENT COUNTER	SDATA
NOAB	I	U	NO ALPHA-BETA CARD FLAG	AERO
NOSCAL	I	U	NO GRID FLAG	PICTUR
NOSPAN	I	U	NUMBER OF ELEMENT DIVISIONS FOR TOP OR BOTTOM	SLABD
NOU	I	U	NUMBER OF U INCREMENTS	ANALY2
NOW	I	U	NUMBER OF W INCREMENTS	ANALY2
NP	I	U	NUMBER OF POINTS TO BE PLOTTED	PLOT
NPCK	I	U	HEADER PRINT CHECK FLAG	FORCE
NPRT	I	U	LINE COUNTER	AERO

SYMBOL	TYPE		DEFINITION	ROUTINE
NPRT	I	U	PRINT COUNTER	FORCE
NPRT	I	U	LINE COUNTER	PICTUR
NPRT	I	U	LINE COUNTER	SDATA
NPRT	I	C	PRINT COUNTER	SKINFR
NPRT	I	C	PRINT COUNTER	TEMP
NPRT	I	U	PRINT LINE COUNTER	VECTOR
NPTS	I	D	NUMBER OF BOUNDARY POINTS	ANALY2
NPT1	I	U	BOUNDARY POINT COUNTER	ANALY2
NPT2	I	U	BOUNDARY POINT COUNTER	ANALY2
NREC	I	U	TAPE 8 RECORD COUNTER	ANALY1
NREC	I	U	NUMBER OF CARDS WRITTEN ON TAPE 8	ANALY2
NREC	I	A	NUMBER OF RECORDS ON TAPE 8	PUNCH
NS	I	U	NUMBER OF SKIN FRICTION SURFACES	AERO
NS	I	A	NUMBER OF SKIN FRICTION SURFACES	FORCE
NS	I	A	NUMBER OF SKIN FRICTION SURFACES	SKINFR
NSAVE	I	U	NUMBER OF SETS OF DATA SAVED FOR SUMMATION	AERO
NSAVE2	I	U	DO-LOOP INDEX FOR DATA SUMMATION	AERO
NSTAT1	I	A	POINT STATUS FLAG	PUNCH
NSTAT2	I	U	POINT STATUS FLAG	PUNCH
NSTAT3	I	A	POINT STATUS FLAG	PUNCH
NU	R	U	FLOW TURNING ANGLE	SHKEXP
NU1	R	U	INITIAL PRANDTL-MEYER ANGLE, RADIANS	EXPAND
NU1D	R	U	INITIAL PRANDTL-MEYER ANGLE, DEGREES	EXPAND
NU2	R	U	FINAL PRANDTL-MEYER ANGLE, RADIANS	EXPAND
NU2D	R	U	FINAL PRANDTL-MEYER ANGLE, DEGREES	EXPAND
NW	R	U	WALL TEMPERATURE CALCULATION FLAG FOR FLOSEP	AERO
NW	R	A	WALL TEMPERATURE CALCULATION FLAG FOR FLOSEP	FLOSEP
NW	R	A	WALL TEMPERATURE CALCULATION FLAG FOR FLOSEP	FORCE
NW	I	A	TEMPERATURE CALCULATION CONTROL FLAG	TEMP
NWI	I	U	TEMPERATURE CALCULATION CONTROL FLAG	TEMP
NWI	I	U	TEMPERATURE CALCULATION PRINT CONTROL FLAG	TEMP
NX	R	U	ELEMENT DIRECTION COSINE-X	CONTRL
NX	R	A	X-COMPONENT OF OUTWARD NORMAL	FLOSEP
NX	R	U	ELEMENT DIRECTION COSINE-X	FORCE
NX	R	U	ELEMENT DIRECTION COSINE-X	PICTUR
NX	I	U	SUBSCRIPT INCREMENT OF X-ARRAY DATA TO BE PLOTTED	PLOT

SYMBOL	TYPE		DEFINITION	ROUTINE
NX	R	U	ELEMENT DIRECTION COSINE-X	SDATA
NX	R	A	ELEMENT DIRECTION COSINE-X	SHKEXP
NX	R	U	FORCE VECTOR DIRECTION COSINE IN X-DIRECTION	VECTOR
NXD	R	U	ELEMENT DIRECTION COSINE-X (CONTROL DEFLECTED)	CONTRL
NXF	R	U	X-COMPONENT OF FLAP SURFACE NORMAL	FLOSEP
NXFS	R	U	X-COMPONENT OF FLAP OUTWARD NORMAL TO BE SAVED	FLOSEP
NXG	I	U	NUMBER OF CHARACTERS IN X-SCALE NUMBER LABELS	PICTUR
NXH	R	U	X-COMPONENT OF OUTWARD NORMAL AT HINGE LINE ELEMENT	FLOSEP
NXI	R	D	ELEMENT DIRECTION COSINE-X	SHKEXP
NXO	R	U	DIRECTION COSINE OUT OF PLANE OF PAPER	PICTUR
NXS	R	U	X-COMPONENT OF SURFACE NORMAL TO BE SAVED	FLOSEP
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	AERO
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY -X	COMPR
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	CONE
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	CONTRL
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	DELWNG
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	EXPAND
NX2	R	C	NX2(1) AND NX2(2) ARE HINGE LINE X-COORDINATE DATA	FLOSEP
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	FORCE
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	NEWTPM
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	QC
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	SDATA
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	SHKEXP
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	SKINFR
NX2	R	C	ELEMENT DIRECTION COSINE ARRAY-X	TEMP
NY	R	U	ELEMENT DIRECTION COSINE-Y	CONTRL
NY	R	A	Y-COMPONENT OF OUTWARD NORMAL	FLOSEP
NY	R	U	ELEMENT DIRECTION COSINE-Y	FORCE
NY	R	U	ELEMENT DIRECTION COSINE-Y	PICTUR
NY	I	U	SUBSCRIPT INCREMENT OF X-ARRAY DATA TO BE PLOTTED.	PLOT
NY	R	U	ELEMENT DIRECTION COSINE-Y	SDATA
NY	R	A	ELEMENT DIRECTION COSINE-Y	SHKEXP
NY	R	U	FORCE VECTOR DIRECTION COSINE IN Y-DIRECTION	VECTOR
NYD	R	U	ELEMENT DIRECTION COSINE-Y (CONTROL DEFLECTED)	CONTRL
NYF	R	U	Y-COMPONENT OF FLAP OUTWARD NORMAL	FLOSEP
NYFS	R	U	Y-COMPONENT OF FLAP OUTWARD NORMAL TO BE SAVED	FLOSEP

SYMBOL	TYPE	DEFINITION	ROUTINE
NYG	I U	NUMBER OF CHARACTERS IN Y-SCALE NUMBER LABELS	PICTUR
NYH	R U	Y-COMPONENT OF OUTWARD NORMAL AT HINGE LINE ELEMENT	FLOSEP
NYI	R D	ELEMENT DIRECTION COSINE-Y	SHKEXP
NYS	R U	Y-COMPONENT OF OUTWARD NORMAL TO BE SAVED	FLOSEP
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	AERO
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	COMPR
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	CONE
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	CONTRL
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	DELWNG
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	EXPAND
NY2	R C	NY2(1) AND NY2(2) ARE HINGE LINE Y-COORDINATE DATA	FLOSEP
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	FORCE
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	NEWTPM
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	QC
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	SDATA
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	SHKEXP
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	SKINFR
NY2	R C	ELEMENT DIRECTION COSINE ARRAY-Y	TEMP
NZ	R U	ELEMENT DIRECTION COSINE-Z	CONTRL
NZ	R A	Z-COMPONENT OF OUTWARD NORMAL	FLOSEP
NZ	R U	ELEMENT DIRECTION COSINE-Z	FORCE
NZ	R U	ELEMENT DIRECTION COSINE-Z	PICTUR
NZ	R U	ELEMENT DIRECTION COSINE-Z	SDATA
NZ	R A	ELEMENT DIRECTION COSINE-Z	SHKEXP
NZ	R U	FORCE VECTOR DIRECTION COSINE IN Z-DIRECTION	VECTOR
NZD	R U	ELEMENT DIRECTION COSINE-Z (CONTROL DEFLECTED)	CONTRL
NZF	R U	Z-COMPONENT OF FLAP SURFACE NORMAL (UNDEFLECTED)	FLOSEP
NZFS	R U	Z-COMPONENT OF FLAP SURFACE NORMAL TO BE SAVED (UNDEFLECTED)	FLOSEP
NZH	R U	Z-COMPONENT OF OUTWARD NORMAL AT HINGE LINE ELEMENT	FLOSEP
NZI	R D	ELEMENT DIRECTION COSINE-Z	SHKEXP
NZS	R U	Z-COMPONENT OF SURFACE NORMAL TO BE SAVED	FLOSEP
NZ2	R C	ELEMENT DIRECTION COSINE ARRAY-Z	AERO
NZ2	R C	ELEMENT DIRECTION COSINE ARRAY-Z	COMPR
NZ2	R C	ELEMENT DIRECTION COSINE ARRAY-Z	CONE
NZ2	R C	ELEMENT DIRECTION COSINE ARRAY-Z	CONTRL
NZ2	R C	ELEMENT DIRECTION COSINE ARRAY-Z	DELWNG

SYMBOL	TYPE		DEFINITION	ROUTINE
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	EXPAND
NZ2	R	C	(NOT USED)	FLOSEP
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	FORCE
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	NEWTPM
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	QC
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	SDATA
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	SHKEXP
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	SKINFR
NZ2	R	C	ELEMENT DIRECTION COSINE ARRAY-Z	TEMP
N1	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
N1	I	U	INDEX	TTABLE
N2	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
N2	I	U	INDEX	TTABLE
ODK	R	U	ODD EXPONENTIAL CONSTANT	BLUNT
ONEOM	R	U	SINE OF ANGLE 3	EXPAND
ONEOM	R	U	1.0 DIVIDED BY MACH NUMBER	SKINFR
P	R	U	FINAL PRESSURE ON ELEMENT WITH SEPARATION	FLOSEP
P	R	U	ROLL RATE	FORCE
P	R	U	PRESSURE	NEWTPM
PAGE	I	C	PAGE NUMBER	AERO
PAGE	I	C	PAGE NUMBER	ANALY1
PAGE	I	C	PAGE NUMBER	ANALY2
PAGE	I	C	PAGE NUMBER	COMPR
PAGE	I	C	PAGE NUMBER	CONE
PAGE	I	C	PAGE NUMBER	CONTRL
PAGE	I	C	PAGE NUMBER	DELWNG
PAGE	I	C	PAGE NUMBER	EXPAND
PAGE	I	C	PAGE NUMBER	FLOSEP
PAGE	I	C	PAGE NUMBER	FORCE
PAGE	I	C	PAGE NUMBER	GRAPIC
PAGE	I	C	PAGE NUMBER	HEADER
PAGE	I	C	PAGE NUMBER	HEADR2
PAGE	I	C	PAGE NUMBER	MAIN
PAGE	I	C	PAGE NUMBER	NEWTPM
PAGE	I	C	PAGE NUMBER	PICTUR
PAGE	I	C	PAGE NUMBER	PLOT

SYMBOL	TYPE		DEFINITION	ROUTINE
PAGE	I	C	PAGE NUMBER	PLUNGE
PAGE	I	C	PAGE NUMBER	PUNCH
PAGE	I	C	PAGE NUMBER	QC
PAGE	I	C	PAGE NUMBER	SDATA
PAGE	I	C	PAGE NUMBER	SHKEXP
PAGE	I	C	PAGE NUMBER	SKINFR
PAGE	I	C	PAGE NUMBER	SLABD
PAGE	I	C	PAGE NUMBER	TEMP
PAGE	I	C	PAGE NUMBER	VECTOR
PBAR	R	U	PRESSURE RATIO (PW/PREF)	ROMU
PBARLG	R	U	LOG10 OF PBAR	ROMU
PBAR1	R	U	LOG10 OF FIRST REFERENCE PRESSURE RATIO	ROMU
PBAR2	R	U	LOG10 OF SECOND REFERENCE PRESSURE RATIO	ROMU
PBAR3	R	U	LOG10 OF THIRD REFERENCE PRESSURE RATIO	ROMU
PC	R	U	FREE-STREAM STATIC TO STAGNATION PRESSURE RATIO	NEWTPM
PCAP	R	U	FREE-STREAM STATIC TO STAGNATION PRESSURE RATIO	NEWTPM
PD	R	U	CORNER POINT PROJECTION DISTANCE	PICTUR
PD	R	U	CORNER POINT PROJECTION DISTANCE	SDATA
PFS	R	U	FREE-STREAM PRESSURE-LBS/SQUARE FOOT	AERO
PFS	R	A	FREE-STREAM PRESSURE	BLUNT
PFS	R	A	FREE-STREAM PRESSURE-LB/FT SQUARED	COMPR
PFS	R	A	FREE-STREAM PRESSURE	FLOSEP
PFS	R	A	FREE-STREAM PRESSURE-LBS / SQUARE FOOT	FORCE
PFS	R	A	FREE-STREAM PRESSURE, LBS / SQUARE FOOT	NEWTPM
PFS	R	A	FREE-STREAM PRESSURE, LBS/SQUARE FOOT	SHKEXP
PFS	R	A	FREE-STREAM PRESSURE-LBS/SQUARE FOOT	SKINFR
PFS	R	A	FREE-STREAM PRESSURE, LBS/SQUARE FOOT	VECTOR
PH	R	U	HINGE LINE INVISCID SHOCK-EXPANSION PRESSURE	FLOSEP
PHI	R	U	COORDINATE TRANSFORMATION ANGLE, DEGREES	CONTRL
PHI	R	U	ANGLE ASSOCIATED WITH GEOMETRY OF SEPARATION	FLOSEP
PHI	R	U	ROLL ANGLE, DEGREES	PICTUR
PHIR	R	U	COORDINATE TRANSFORMATION ANGLE, RADIANS	CONTRL
PHIR	R	U	ROLL ANGLE, RADIANS	FORCE
PI	R	U	RATIO OF CIRCUMFERENCE OF A CIRCLE TO ITS DIAMETER	PLUNGE
PI	R	D	PRESSURE	SHKEXP
PM	R	D	MATRIX OF ATMOSPHERIC PRESSURES	ATMOS

SYMBOL	TYPE		DEFINITION	ROUTINE
PNIN	R	U	INVISCID PRESSURE USING NORMAL FORCE METHOD	FLOSEP
PO	R	U	PRESSURE RATIO AT LAMBDA = 0.0	SKINFR
POLY	R	U	VALUE OF POLYNOMIAL	POLY
PPFS	R	U	LOCAL TO FREE-STREAM PRESSURE RATIO	NEWTPM
PPO	R	U	SURFACE PRESSURE RATIO	NEWTPM
PPPO	R	U	RATIO OF PLATEAU PRESSURE TO FREE-STREAM PRESSURE	FLOSEP
PPPUX	R	U	PLATEAU PRESSURE/STREAM PRESSURE AT SEPARATION POINT	FLOSEP
PPPO1	R	U	PLATEAU PRESS/FREE-STREAM PRESS ON ELEMENT BEFORE SEPARATION	FLOSEP
PPT2	R	U	OSU METHOD PRESSURE RATIO	FORCE
PRAN	R	U	PRANDTL NUMBER	TEMP
PREF	R	U	REFERENCE PRESSURE (2117.36 LB/SQ.FT.)	ROMU
PRINT	I	U	DETAIL FORCE CONTRIBUTION PRINT FLAG	AERO
PRINT	I	A	PRINT FLAG	FORCE
PRINT	R	D	PRINTING ARRAY	PLOT
PRINTS	I	U	PRINT FLAG FOR ELEMENT DATA	AERO
PRINTS	I	U	ELEMENT DATA PRINT FLAG	PICTUR
PRINTS	I	A	ELEMENT DATA PRINT FLAG	SDATA
PSEIN	R	D	INVISCID SHOCK-EXPANSION PRESSURE	FLOSEP
PSEVIS	R	D	SHOCK-EXPANSION PRESSURE WITH VISCOUS SEPARATION	FLOSEP
PSI	R	U	COORDINATE TRANSFORMATION ANGLE, DEGREES	CONTRL
PSI	R	U	YAW ANGLE	PICTUR
PSIR	R	U	COORDINATE TRANSFORMATION ANGLE, RADIANS	CONTRL
PSPAN	R	U	PER CENT SEMI-SPAN	SLABD
PSTAG	R	U	WIND TUNNEL STAGNATION PRESSURE, ATMOSPHERES	AERO
PSTAG	R	A	WIND TUNNEL STAGNATION PRESSURE-LBS / SQUARE FOOT	FORCE
PT2PO	R	U	OSU METHOD PRESSURE RATIO BEHIND NORMAL SHOCK	FORCE
PW	R	U	PRESSURE	QC
PW	R	A	LOCAL PRESSURE (LB/SQ.FT.)	ROMU
PX	R	U	LOCAL PRESSURE AT EXACT SEPARATION POINT	FLOSEP
PXOD	R	U	INCREMENT FROM ORIGIN, ODD EXPONENTIAL	BLUNT
PX1	R	U	PRESSURE ON ELEMENT JUST BEFORE SEPARATION POINT	FLOSEP
P1	R	U	FIRST ITERATION PRESSURE	NEWTPM
P2	R	U	SECOND ITERATION PRESSURE	NEWTPM
P2	R	A	LOCAL PRESSURE (LB/SQ.FT.)	ROMU
P2P11	R	U	PRESSURE RATIO ACROSS COMPRESSION	CONE
P2P11	R	U	PRESSURE RATIO	SHKEXP

SYMBOL	TYPE		DEFINITION	ROUTINE
Q	R	U	PITCH RATE, RADIANS / SECOND	FORCE
Q	R	U	MATCHING POINT TO FREE-STREAM STATIC PRESSURE RATIO	NEWTPM
Q	R	U	TAIL EFFECTIVENESS RATIO	PLUNGE
Q	R	U	TAPER RATIO CORRECTION EQUATION EXPONENT	SKINFR
Q	R	U	DYNAMIC PRESSURE	VECTOR
QC	R	U	CONVECTIVE HEATING RATE	QC
QC1	R	U	CONVECTIVE HEATING RATE AT TC1	TEMP
QC2	R	U	CONVECTIVE HEATING RATE AT TC2	TEMP
QQINF	R	U	DYNAMIC PRESSURE RATIO CORRECTION FACTOR	AERO
QQINF	R	A	DYNAMIC PRESSURE RATIO CORRECTION FACTOR	FORCE
QQINFS	R	C	SAVED VALUES OF DYNAMIC PRESSURE CORRECTION	AERO
QQINFS	R	C	SAVED VALUES OF DYNAMIC PRESSURE CORRECTION	FORCE
QRP	R	U	INPUT VEHICLE ROTATION RATE, RADIANS/SECOND	AERO
QRP	R	A	INPUT VEHICLE ROTATION RATE, RADIANS / SECOND	FORCE
QRPS	R	D	SAVED VALUES OF ROTATION RATE	AERO
QRPSS	R	U	SAVED VALUE OF INPUT ROTATION RATE	AERO
QR1	R	U	RADIATION HEATING RATE AT TR1	TEMP
QR2	R	U	RADIATION HEATING RATE AT TR1	TEMP
Q1	R	A	QUADRATIC INTERPOLATION DATA ARRAY	QINT
Q1	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q10	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q11	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q2	R	A	QUADRATIC INTERPOLATION DATA ARRAY	QINT
Q2	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q3	R	A	QUADRATIC INTERPOLATION DATA ARRAY	QINT
Q3	R	U	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q4	R	A	QUADRATIC INTERPOLATION DATA ARRAY	QINT
Q4	R	U	QUADRATIC INTERPOLATION VARIABLE	TTABLE
Q5	R	D	QUADRATIC INTERPOLATION DATA ARRAY	QINT
Q8	R	D	QUADRATIC INTERPOLATION VARIABLE	TTABLE
R	R	D	ROOTS OF CUBIC EQUATION	COMPR
R	R	U	VARIABLE IN MACH NUMBER EQUATION	EXPAND
R	R	U	YAW RATE, RADIANS / SECOND	FORCE
R	R	U	BODY RADIUS AT WING OR TAIL	PLUNGE
R	R	U	LOCAL RADIUS	SLABD
R	R	A	DATA ARRAY	TTABLE

SYMBOL	TYPE		DEFINITION	ROUTINE
RA	R	U	REYNOLDS ANALOGY FACTOR	QC
RAD	R	U	ELLIPSE RADIUS	ANALY1
RADIUS	R	U	RADIUS	SLABD
RADK	R	U	RADIATION CONSTANT	TEMP
RATIO	R	U	DUMMY VARIABLE	PLUNGE
RB	R	A	BODY NOSE RADIUS (FEET)	BLUNT
RE	R	D	REFERENCE REYNOLDS NUMBER	FLOSEP
RE	R	U	REYNOLDS NUMBER	NEWTPM
RE	R	D	REFERENCE REYNOLDS NUMBER	SKINFR
RE	T	A	REFERENCE REYNOLDS NUMBER	TEMP
REAFT	R	U	LOCAL SURFACE REYNOLDS NUMBER PER FOOT	FLOSEP
REAHL	R	U	REYNOLDS NUMBER AT HINGE LINE ELEMENT	FLOSEP
REASFT	R	U	REFERENCE REYNOLDS NUMBER PER FOOT	FLOSEP
REAXOP	R	U	REYNOLDS NUMBER ON LOCAL ELEMENT	FLOSEP
RELOC	R	U	LOCAL REYNOLDS NUMBER	SKINFR
RENO	R	U	FREE STREAM REYNOLDS NUMBER	AERO
RENO	R	A	FREE-STREAM REYNOLDS NUMBER PER FOOT	BLUNT
RENO	R	A	FREE STREAM REYNOLDS NUMBER	FORCE
RENO	R	A	FREE-STREAM REYNOLDS NUMBER	SHKEXP
RENO	R	U	FREE-STREAM REYNOLDS NUMBER	SKINFR
RES	R	U	SHOCK REYNOLDS NUMBER	BLUNT
RET	R	C	TURBULENT FLOW REYNOLDS NUMBER AT REFERENCE CONDITION	QC
RET	R	C	TURBULENT FLOW REYNOLDS NUMBER AT REFERENCE CONDITION	SKINFR
RET	R	C	TURBULENT FLOW REYNOLDS NUMBER AT REFERENCE CONDITION	TEMP
RETRAN	R	U	TRANSITION REYNOLDS NUMBER FOR CONTROL SURFACE	AERO
RETRAN	R	A	INPUT FLOW TRANSITION REYNOLD'S NUMBER	FLOSEP
RETRAN	R	A	TRANSITION REYNOLDS NUMBER FOR CONTROL SURFACE	FORCE
RF	R	D	RECOVERY FACTOR	TEMP
RFLAG	L	U	INPUT DATA READ CONTROL FLAG	PICTUR
RFLAG	L	U	INPUT DATA READ CONTROL FLAG	SDATA
RHO	R	U	DENSITY	NEWTPM
RHOFS	R	A	FREE-STREAM DENSITY	BLUNT
RHOFS	R	U	FREE STREAM DENSITY	FORCE
RHOFS	R	A	FREE-STREAM DENSITY	SHKEXP
RHOFS	R	A	FREE STREAM DENSITY	SKINFR
RNOSE	R	U	RADIUS OF NOSE AND LEADING EDGE	SLABD

SYMBOL	TYPE		DEFINITION	ROUTINE
ROLL	R	U	ROLL ANGLE, DEGREES	FORCE
ROLLR	R	U	ROLL ANGLE, RADIANS	FORCF
ROLLR	R	U	ROLL ANGLE IN RADIANS	VECTOR
ROMU	R	U	DENSITY-VISCOSITY PRODUCT	ROMU
ROMURA	R	C	SQUARE-ROOT OF REFERENCE DENSITY-VISCOSITY RATIO	QC
ROMURA	R	C	SQUARE-ROOT OF REFERENCE DENSITY-VISCOSITY RATIO	SKINFR
ROMURA	R	C	SQUARE-ROOT OF REFERENCE DENSITY-VISCOSITY RATIO	TEMP
RORA	R	U	REFERENCE TO FREE-STREAM DENSITY-VISCOSITY RATIO	TEMP
RORAI	R	U	INVERSE DENSITY RATIO ACROSS NORMAL SHOCK	BLUNT
ROSTAR	R	U	DENSITY AT REFERENCE CONDITION	QC
ROW	R	U	ENTRY TO DETERMINE DENSITY (ALSO DENSITY PARAMETER)	ROMU
ROWS	I	U	NOT USED	SLABD
RS	R	U	R DIVIDED BY S	PLUNGE
RT	R	U	RECOVERY TEMPERATURE	FLOSEP
RT	R	D	RECOVERY TEMPERATURE	SKINFR
RT	R	A	RECOVERY TEMPERATURE	TEMP
RO	R	U	EARTH RADIUS = 20890855 FEET	ATMOS
R3	R	U	SHOCK ANGLE PARAMETER	CONE
R3	R	U	PARAMETER IN TEMPERATURE EQUATION	DELWNG
S	R	U	BOUNDARY-LENGTH	ANALY2
S	R	U	FREE MOLECULAR FLOW SPEED RATIO	FORCE
S	R	U	WING/TAIL SEMI-SPAN	PLUNGE
SCF	R	D	TOTAL SKIN FRICTION	SKINFR
SCFA	R	D	TOTAL SKIN FRICTION	SKINFR
SDELTQ	R	U	SINE OF MATCHING POINT IMPACT ANGLE	NEWTPM
SECT	R	D	SECTION IDENTIFICATION	ANALY1
SECT	R	D	SECTION IDENTIFICATION	ANALY2
SECT	R	A	SECTION IDENTIFICATION	PUNCH
SECT	R	U	SECTION IDENTIFICATION	SDATA
SECTS	R	U	SECTION IDENTIFICATION	SDATA
SEQ	I	U	CARD SEQUENCE NUMBER	ANALY1
SEQ	I	U	CARD SEQUENCE NUMBER	ANALY2
SEQ	I	A	CARD SEQUENCE NUMBER	PUNCH
SEQ	I	U	CARD SEQUENCE NUMBER	SDATA
SEQ	I	U	CARD SEQUENCE NUMBER	SLABD
SFRONT	R	U	BODY FRONTAL AREA	PLUNGE

SYMBOL	TYPE		DEFINITION	ROUTINE
SHEAR	R	U	FREE MOLECULAR FLOW SHEAR FORCE	FORCE
SHEARX	R	U	X-COMPONENT OF FREE MOLECULAR FLOW SHEAR FORCE	FORCE
SHEARY	R	U	Y-COMPONENT OF FREE MOLECULAR FLOW SHEAR FORCE	FORCE
SHEARZ	R	U	Z-COMPONENT OF FREE MOLECULAR FLOW SHEAR FORCE	FORCE
SINDE	R	U	SINE OF CONTROL DEFLECTION ANGLE	CONTRL
SINNU	R	U	SINE OF FLOW TURNING ANGLE	SHKEXP
SINPHI	R	U	SINE OF PHI	CONTRL
SINPHI	R	U	SIN OF PHI	PICTUR
SINPSI	R	U	SINE OF PSI	CONTRL
SINPSI	R	U	SIN OF PSI	PICTUR
SINTH	R	U	SINE OF SHOCK ANGLE	FLOSEP
SINTH	R	U	SIN OF THETA	PICTUR
SKIN	R	U	TOTAL AXIAL FORCE SKIN FRICTION CONTRIBUTION	FORCE
SKIN	R	A	TOTAL AXIAL SKIN FRICTION CONTRIBUTION	SKINFR
SPAN	R	U	REFERENCE LENGTH FOR ROLLING, YAWING COEFFICIENTS	AERO
SPAN	R	A	REFERENCE LENGTH FOR ROLLING, YAWING COEFFICIENTS	FORCE
SPAN	R	A	REFERENCE LENGTH FOR ROLLING, YAWING COEFFICIENTS	VECTOR
SR	R	U	S DIVIDED BY R	PLUNGE
SREF	R	U	VEHICLE REFERENCE AREA (WING AREA)	AERO
SREF	R	A	VEHICLE REFERENCE AREA (WING AREA)	FORCE
SREF	R	A	VEHICLE REFERENCE AREA (WING AREA)	SKINFR
SREF	R	A	VEHICLE REFERENCE AREA (WING AREA)	VECTOR
SSIND	R	U	SPEED RATIO TIMES SINE OF IMPACT ANGLE	FORCE
STAT	I	U	POINT STATUS FLAG	ANALY1
STAT	I	U	SURFACE POINT STATUS FLAG	ANALY2
STAT	I	U	COORDINATE POINT STATUS FLAG	PICTUR
STAT	I	U	COORDINATE POINT STATUS FLAG	SDATA
STAT	I	U	COORDINATE POINT STATUS FLAG	SLABD
STATA	I	U	POINT STATUS FLAG	ANALY1
STATA	I	U	COORDINATE POINT STATUS FLAG	SLABD
STATB	I	U	POINT STATUS FLAG	ANALY1
STATC	I	U	POINT STATUS FLAG	ANALY1
STATD	I	U	POINT STATUS FLAG	ANALY1
STATT	I	U	POINT STATUS FLAG	ANALY1
STATT	I	U	SURFACE POINT STATUS FLAG	ANALY2

SYMBOL	TYPE		DEFINITION	ROUTINE
STATZ	I	U	COORDINATE POINT STATUS FLAG	PICTUR
STATT	I	U	COORDINATE POINT STATUS FLAG	SDATA
STATY	I	U	COORDINATE POINT STATUS FLAG	SLABD
STOTAL	R	U	TOTAL VALUE OF SHEAR FORCE VECTOR	FORCE
SURF	R	C	SKIN FRICTION DATA ARRAY	AERO
SURF	R	C	SKIN FRICTION DATA ARRAY	FORCE
SURF	R	C	SKIN FRICTION DATA ARRAY	SKINFR
SURF	R	C	SKIN FRICTION DATA ARRAY	TEMP
SWBVS	R	U	WING/TAIL AREA DIVIDED BY REFERENCE AREA	PLUNGE
SWEEP	R	U	LEADING EDGE SWEEP (NOT USED BY MARK II)	AERO
SWEEP	R	A	LEADING EDGE SWEEP ANGLE	FLOSEP
SWEEP	R	A	LEADING EDGE SWEEP (NOT USED BY MARK II)	FORCE
SWEEP	R	U	LEADING EDGE SWEEP ANGLE	SLABD
SX	R	U	SHEAR FORCE VECTOR COMPONENT-X	FORCE
SY	R	U	SHEAR FORCE VECTOR COMPONENT-Y	FORCE
SYMFCO	I	U	SYMMETRY FACTOR	FORCE
SYMFCT	I	U	SYMMETRY FLAG	AERO
SYMFCT	I	A	SYMMETRY FACTOR	FORCE
SYMFCT	I	U	SYMMETRY FLAG	PICTUR
SYMFCT	I	A	SYMMETRY FLAG	SDATA
SZ	R	U	SHEAR FORCE VECTOR COMPONENT-Z	FORCE
T	R	U	TEMPERATURE	NEWTPM
T	R	U	UNIT VECTOR	PICTUR
T	R	U	UNIT VECTOR	SDATA
TANPHI	R	U	TANGENT OF FLOW SEPARATION ANGLE	FLOSEP
TAPER1	R	U	TAPER RATIO OF INITIAL SURFACE	SKINFR
TAPER2	R	U	TAPER RATIO OF SURFACE	SKINFR
TAU	R	A	SHEAR FORCE	BLUNT
TBTIN	R	U	RATIO OF BODY TEMPERATURE TO FREE-STREAM TEMPERATURE	FORCE
TCT1	R	U	SUTHERLAND CONSTANT TO FREE-STREAM TEMPERATURE RATIO	QC
TC1	R	U	FIRST VALUE OF CONVECTIVE TEMPERATURE	TEMP
TC2	R	U	SECOND VALUE OF CONVECTIVE TEMPERATURE	TEMP
TEXT	I	D	INFORMATION ON TYPE 3 CARD FROM CC 1 TO CC 71	CARD
TFS	R	A	FREE-STREAM TEMPERATURE	BLUNT
TFS	R	A	FREE-STREAM TEMPERATURE-DEGREE R	COMPR
TFS	R	A	FREE-STREAM TEMPERATURE	FLOSEP
TFS	R	U	FREE-STREAM TEMPERATURE	FORCE

SYMBOL	TYPE		DEFINITION	ROUTINE
TFS	R	A	FREE STREAM TEMPERATURE	NEWTPH
TFS	R	A	FREE STREAM TEMPERATURE	SHKEXP
TFS	R	A	FREE-STREAM TEMPERATURE	SKINFR
TH	R	U	FLOW TEMPERATURE AT HINGE LINE ELEMENT	FLOSEP
THCHK	R	U	ANGULAR CHECK PARAMETER	SLABD
THEMAX	R	U	MAXIMUM VALUE OF THETA	SLABD
THESID	R	U	THETA AT THE SIDE	SLABD
THETA	R	U	ANGULAR POSITION	ANALY1
THETA	R	U	SHOCK ANGLE	FLOSEP
THETA	R	U	SURFACE SLOPE	FORCE
THETA	R	U	PITCH ANGLE	PICTUR
THETA	R	U	ANGULAR POSITION IN Z-Y PLANE (FROM BOTTOM)	SLABD
THETAB	I	U	NUMBER OF ANGULAR DIVISIONS ON THE BOTTOM	SLABD
THETAO	R	U	STARTING VALUE OF THETA	SLABD
THETAP	R	U	ANGULAR POSITION	ANALY1
THETAT	I	U	NUMBER OF ANGULAR DIVISIONS ON THE TOP	SLABD
THETA2	R	U	ANGULAR POSITION	SLABD
THETL	R	U	LAST THETA ANGLE	ANALY1
THETLX	R	D	LAST THETA ANGLE ARRAY	ANALY1
THETO	R	U	INITIAL THETA	ANALY1
THETOX	R	D	INITIAL THETA ANGLE ARRAY	ANALY1
TI	R	D	LOCAL TEMPERATURE	SHKEXP
TINT2	R	U	TEMPERATURE RATIO	CONE
TINT2	R	U	TEMPERATURE PARAMETER FOR CONE MACH NUMBER EQUATION	FORCE
TINT2	R	U	TEMPERATURE RATIO	SHKEXP
TITLE	R	C	TITLE ARRAY	AERO
TITLE	R	C	TITLE	ANALY1
TITLE	R	C	TITLE	ANALY2
TITLE	R	C	TITLE	COMPR
TITLE	R	C	TITLE	CONE
TITLE	R	C	TITLE	CONTRL
TITLE	R	C	TITLE	DELWNG
TITLE	R	C	TITLE	EXPAND
TITLE	R	C	TITLE ARRAY	FLOSEP
TITLE	R	C	TITLE	FORCE
TITLE	R	C	TITLE	GRAPIC

SYMBOL	TYPE	DEFINITION	ROUTINE
TITLE	R C	TITLE	HEADER
TITLE	R C	TITLE	HEADR2
TITLE	R C	PROBLEM TITLE	MAIN
TITLE	R C	TITLE	NEWTPM
TITLE	R C	TITLE	PICTUR
TITLE	R D	ABSISSA AND ORDINATE TITLES, AND HORIZONTAL TITLE ARRAY	PLOT
TITLE	R C	TITLE	PLUNGE
TITLE	R C	TITLE	PUNCH
TITLE	R C	TITLE	QC
TITLE	R C	TITLE	SDATA
TITLE	R C	TITLE	SHKEXP
TITLE	R C	TITLE	SKINFR
TITLE	R C	TITLE	SLABD
TITLE	R C	TITLE	TEMP
TITLE	R C	TITLE	VECTOR
TITLE2	R C	DUMMY TITLE ARRAY	PLOT
TM	R D	MATRIX OF MOLECULAR SCALE TEMPERATURES, DEG RANKINE	ATMOS
TMS	R U	MOLECULAR SCALE TEMPERATURE, DEGREE RANKINE	ATMOS
TOPTC	R U	TOP THICKNESS CORRECTION FACTOR	SLABD
TR	R D	TEMPERATURE DATA ARRAY	FLOSEP
TR	R D	FLIGHT CONDITION AND SKIN FRICTION DATA ARRAY	SKINFR
TR	R A	FLIGHT CONDITION AND SKIN FRICTION DATA ARRAY	TEMP
TR1	R U	FIRST VALUE OF RADIATION TEMPERATURE	TEMP
TR2	R U	SECOND VALUE OF RADIATION TEMPERATURE	TEMP
TS	R D	REFERENCE TEMPERATURE (DEGREE R)	FLOSEP
TS	R D	REFERENCE TEMPERATURE (T STAR)	SKINFR
TS	R A	REFERENCE TEMPERATURE (T STAR)	TEMP
TSTAG	R U	WIND TUNNEL STAGNATION TEMPERATURE-DEGREES F	AERO
TSTAG	R A	WIND TUNNEL STAGNATION TEMPERATURE, DEGREES F	FORCE
TSTAR	R U	REFERENCE TEMPERATURE	QC
TST1	R C	REFERENCE TO FREE-STREAM TEMPERATURE (OR ENTHALPY) RATIO	QC
TST1	R C	REFERENCE TO FREE-STREAM TEMPERATURE (OR ENTHALPY) RATIO	SKINFR
TST1	R C	REFERENCE TO FREE-STREAM TEMPERATURE (OR ENTHALPY) RATIO	TEMP
TSUBT	R U	FREE STREAM TOTAL TEMPERATURE	NEWTPM
TW	R A	WALL TEMPERATURE	QC
TW	R A	TEMPERATURE (RANKINE)	ROMU

SYMBOL	TYPE	DEFINITION	ROUTINE
TWALL	R U	WALL TEMPERATURE FOR FLOSEP	AERO
TWALL	R A	WALL TEMPERATURE FOR FLOSEP	FLOSEP
TWALL	R A	WALL TEMPERATURE FOR FLOSEP	FORCE
TWBL	R U	RATIO OF SHEAR FORCE WITH INTERACTION TO THAT WITHOUT	BLUNT
TWTL	R U	WALL TO FREE-STREAM TEMPERATURE RATIO, LAMINAR	SKINFR
TWTRL	R U	WALL TO RECOVERY TEMPERATURE RATIO, LAMINAR	SKINFR
TWX	R U	REDUCED TEMPERATURE (TW*1.0E-4)	ROMU
TW1	R U	WALL TEMPERATURE	QC
TX	R U	TANGENT VECTOR X-COMPONENT	ANALY2
TX	R U	FREE MOLECULAR FLOW VECTOR COMPONENT-X	FORCE
TX1	R U	FLOW TEMPERATURE ON ELEMENT JUST BEFORE SEPARATION ELEMENT	FLOSEP
TY	R U	TANGENT VECTOR Y-COMPONENT	ANALY2
TY	R U	FREE MOLECULAR FLOW VECTOR COMPONENT-Y	FORCE
TYPE	I U	CARD TYPE	AERO
TYPE	I U	CARD TYPE FOR GEOMETRY DATA =3	ANALY1
TYPE	I U	CARD TYPE NUMBER	ANALY2
TYPE	I U	CARD TYPE	CARD
TYPE	I U	CARD TYPE	MAIN
TYPE	I U	CARD TYPE NUMBER	PICTUR
TYPE	I U	CARD TYPE	PLOT
TYPE	I U	CARD TYPE	PLUNGE
TYPE	I A	CARD TYPE NUMBER	PUNCH
TYPE	I U	CARD TYPE NUMBER	SDATA
TYPE	I U	CARD TYPE NUMBER	SLABD
TYPE	I U	CARD TYPE NUMBER	VECTOR
TZ	R U	TANGENT VECTOR Z-COMPONENT	ANALY2
TZ	R U	FREE MOLECULAR FLOW VECTOR COMPONENT-Z	FORCE
T1	R U	DUMMY VARIABLE	PLUNGE
T1X	R U	X-COMPONENT OF VECTOR T1	PICTUR
T1X	R U	X-COMPONENT OF VECTOR T1	SDATA
T1Y	R U	Y-COMPONENT OF VECTOR T1	PICTUR
T1Y	R U	Y-COMPONENT OF VECTOR T1	SDATA
T1Z	R U	Z-COMPONENT OF VECTOR T1	PICTUR
T1Z	R U	Z-COMPONENT OF VECTOR T1	SDATA
T2	R U	TEMPERATURE BEHIND NORMAL SHOCK	BLUNT
T2	R U	DUMMY VARIABLE	PLUNGE
T2X	R U	X-COMPONENT OF VECTOR T2	PICTUR

SYMBOL	TYPE		DEFINITION	ROUTINE
T2X	R	U	X-COMPONENT OF VECTOR T2	SDATA
T2Y	R	U	Y-COMPONENT OF VECTOR T2	PICTUR
T2Y	R	U	Y-COMPONENT OF VECTOR T2	SDATA
T2Z	R	U	Z-COMPONENT OF VECTOR T2	PICTUR
T2Z	R	U	Z-COMPONENT OF VECTOR T2	SDATA
T3	R	U	DUMMY VARIABLE	PLUNGE
T4	R	U	DUMMY VARIABLE	PLUNGE
U	R	U	PARAMETRIC VARIABLE, U	ANALY2
UPWASH	R	U	TAIL UPWASH DERIVATIVE CAUSED BY WING	PLUNGE
U2	R	U	PARAMETRIC VARIABLE U SQUARED	ANALY2
U3	R	U	PARAMETRIC VARIABLE U CUBED	ANALY2
V	R	U	FREE-STREAM VELOCITY-FEET/SECOND	AERO
V	R	A	FREE-STREAM VELOCITY, FEET / SECOND	FORCE
V	R	U	VELOCITY	NEWTPM
V	R	A	FREE STREAM VELOCITY, FEET/SECOND	SHKEXP
VBAR	R	U	HYPERSONIC VISCOUS PARAMETER	SKINFR
VIS	R	A	FREE-STREAM VISCOSITY	BLUNT
VIS	R	U	FREE-STREAM VISCOSITY	FORCE
VIS	R	U	VICOSITY AT REFERENCE CONDITION	QC
VIS	R	A	FREE-STREAM VISCOSITY	SHKEXP
VIS	R	A	FREE STREAM VISCOSITY	SKINFR
VISRA	R	U	REFERENCE TO FREE-STREAM VISCOSITY RATIO	QC
VISTAR	R	U	VICOSITY AT REFERENCE CONDITION	SKINFR
VISWL	R	U	VISCOSITY AT WALL TEMPERATURE, LAMINAR	SKINFR
VISWT	R	U	VISCOSITY AT WALL TEMPERATURE, TURBULENT	SKINFR
VIS2	R	U	VISCOSITY BEHIND NORMAL SHOCK	BLUNT
VLOCAL	R	U	TOTAL LOCAL VELOCITY	FORCE
VN	R	U	VECTOR LENGTH	PICTUR
VN	R	U	VECTOR LENGTH	SDATA
VOL	R	U	TOTAL VOLUME	PICTUR
VOL	R	U	TOTAL VOLUME	SDATA
VOLUME	R	U	BODY VOLUME	PLUNGE
VSTAR	R	U	VICOUS-INTERACTION PARAMETER	SKINFR
VTITLE	R	D	VERTICAL SCALE TITLE	PICTUR
VX	R	U	LOCAL VELOCITY COMPONENT-X	FORCE
VXI	R	U	FREE-STREAM VELOCITY COMPONENT-X	FORCE

SYMBOL	TYPE		DEFINITION	ROUTINE
VY	R	U	LOCAL VELOCITY COMPONENT-Y	FORCE
VYI	R	U	FREE-STREAM VELOCITY COMPONENT-Y	FORCE
VZ	R	U	LOCAL VELOCITY COMPONENT-Z	FORCE
VZI	R	U	FREE STREAM VELOCITY COMPONENT-Z	FORCE
W	R	U	PARAMETRIC VARIABLE, W	ANALY2
W	R	U	PARAMETER IN CUBIC EQUATION	COMPR
W	R	D	GRID INFORMATION ARRAY	PLOT
WM	R	D	MATRIX OF MOLECULAR WEIGHTS OF AIR	ATMOS
WMO	R	U	MOLECULAR WEIGHT OF AIR AT SEA LEVEL = 28.9644	ATMOS
W2	R	U	PARAMETRIC VARIABLE W SQUARED	ANALY2
W3	R	U	PARAMETRIC VARIABLE W CUBED	ANALY2
X	R	U	X-COORDINATE	ANALY1
X	R	U	X-COORDINATE	ANALY2
X	R	U	SHOCK ANGLE PARAMETER	CONE
X	R	U	SHOCK ANGLE PARAMETER	DELWNG
X	R	U	X-COORDINATE	PICTUR
X	R	D	PLOTTING ARRAY, LOCATION ALONG X-AXIS	PLOT
X	R	U	X-COORDINATE	SDATA
X	P	U	X-COORDINATE	SLABD
XA	R	U	X-COORDINATE	ANALY1
XA	R	D	X-COORDINATE	ANALY2
XA	R	D	X-COORDINATE	PICTUR
XA	R	D	X-COORDINATE	SDATA
XA	R	U	X-COORDINATE	SLABD
XATACH	R	A	X-COORDINATE AT FLOW ATTACHMENT POINT	FLOSEP
XATACH	R	U	X-COORDINATE AT FLOW ATTACHMENT POINT	FORCE
XAVG	R	U	AVERAGE X-COORDINATE	SDATA
XB	R	U	X-COORDINATE	ANALY1
XB	R	D	X-COORDINATE	ANALY2
XB	R	D	X-COORDINATE	PICTUR
XB	R	D	X-COORDINATE	SDATA
XB	R	U	INPUT X-COORDINATE STATION	SLABD
XBTBYC	R	U	TAIL LENGTH DIVIDED BY REFERENCE CHORD	PLUNGE
XB1	R	D	BOUNDARY CURVE X-COORDINATE ARRAY	ANALY2
XC	R	U	X-COORDINATE	ANALY1
XC	R	U	AREA CENTROID LOCATION OF BODY	PLUNGE

SYMBOL	TYPE		DEFINITION	ROUTINE
XCENT	R	U	QUADRILATERAL ELEMENT CENTROID-X	CONTRL
XCENT	R	A	HINGE MOMENT FACTOR FOR CONTROL SURFACE ELEMENT	FLOSEP
XCENT	R	U	QUADRILATERAL ELEMENT CENTROID-X	FORCE
XCENT	R	U	ELEMENT CENTROID COORDINATE-X	PICTUR
XCENT	R	U	ELEMENT CENTROID COORDINATE-X	SDATA
XCENT	R	U	ACTION POINT FOR FORCE VECTOR-X	VECTOR
XCENTD	R	U	QUADRILATERAL ELEMENT CENTROID-X (CONTROL DEFLECEDD)	CONTRL
XCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-X	AERO
XCENT2	R	C	ELEMENT CENTROID ARRAY-X	COMPR
XCENT2	R	C	ELEMENT CENTROID ARRAY-X	CONE
XCENT2	R	C	HINGE MOMENT FACTOR	CONTRL
XCENT2	R	C	ELEMENT CENTROID COORDINATE-X	DELWNG
XCENT2	R	C	ELEMENT CENTROID ARRAY-X	EXPAND
XCENT2	R	C	HINGE MOMENT FACTOR ARRAY FOR CONTROL SURFACE ELEMENTS	FLOSEP
XCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-X	FORCE
XCENT2	R	C	ELEMENT CENTROID COORDINATE-X	NEWTPM
XCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-X	QC
XCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-X	SDATA
XCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-X	SHKEXP
XCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-X	SKINFR
XCENT2	R	C	QUADRILATERAL CENTROID ARRAY-X	TEMP
XCG	R	U	X-CENTER FOR MOMENT CALCULATIONS	AERO
XCG	R	A	X-CENTER FOR MOMENT CALCULATIONS	FORCE
XCG	R	A	X-CENTER FOR MOMENT CALCULATIONS	VECTOR
XD	R	U	X-COORDINATE	ANALYI
XDIF	R	U	COORDINATE DIFFERENCE-X	PICTUR
XDIF	R	U	COORDINATE DIFFERENCE-X	SDATA
XHL	R	U	ELEMENT AVERAGE X-COORDINATE AT HINGE LINE	FLOSEP
XHL1	R	U	HINGE LINE X-COORDINATE OF POINT 1	CONTRL
XHL4	R	U	HINGE LINE X-COORDINATE OF POINT 4	CONTRL
XI	R	D	COORDINATE IN ELEMENT COORDINATE SYSTEM	PICTUR
XI	R	D	COORDINATE IN ELEMENT COORDINATE SYSTEM	SDATA
XIN	R	D	ELEMENT COORDINATES-X	PICTUR
XIN	R	D	ELEMENT COORDINATES-X	SDATA
XIO	R	U	CENTROID IN ELEMENT COORDINATE SYSTEM	PICTUR
XIO	R	U	CENTROID IN ELEMENT COORDINATE SYSTEM	SDATA

SYMBOL	TYPE	DEFINITION	ROUTINE
XI3M1	R U	CONSTANT FOR AREA EQUATION	PICTUR
XI3M1	R U	CONSTANT FOR AREA EQUATION	SDATA
XL	R U	LEFT-MOST LIMIT OF THE GRID ON X-AXIS	PLOT
XLE	R U	DISTANCE FROM LEADING EDGE TO ELEMENT CENTROID	CONTRL
XLE	R A	X-DISTANCE FROM CENTROID OF ELEMENT TO LEADING EDGE LINE	FLOSEP
XLE	R U	X-DISTANCE FROM CENTRAID OF ELEMENT TO LEADING EDGE LINE	FORCE
XLE	R U	DISTANCE FROM LEADING EDGE TO ELEMENT CENTROID	SDATA
XLEH	R U	X-DISTANCE FROM LEADING EDGE TO HINGE LINE	FLOSEP
XLEO	R U	LEADING EDGE X INCREMENT	SDATA
XLEP1	R U	SAVED X-COORDINATE	SDATA
XLESEP	R U	DISTANCE FROM LEADING EDGE TO SEPARATION POINT	FLOSEP
XLE1	R U	ELEMENT CENTROID DISTANCE FROM L. E. BEFORE SEPARATION	FLOSEP
XLG	R U	VALUE OF LEFT SIDE OF HORIZONTAL SCALE	PICTUR
XOEV	R U	ORIGIN FOR EVEN EXPONENTIAL	BLUNT
XOP	R U	X IN TRANSFORMED SYSTEM	CONTRL
XOPDE	R U	X IN TRANSFORMED SYSTEM (CONTROL DEFLECTED)	CONTRL
XOPH	R U	X-COORDINATE	CONTRL
XOW	R U	BOUNDARY CURVE POINT, X(O,W)	ANALY2
XP	R U	X-COORDINATE	CONTRL
XPA	R D	COORDINATES OF ELEMENT CORNER POINTS, X	CONTRL
XPA	R A	X-COORDINATES OF QUADRILATERAL ELEMENT	FLOSEP
XPA	R D	X-COORDINATES OF QUADRILATERAL ELEMENT	FORCE
XPA	R U	COORDINATE OF ELEMENT CORNER POINT	PICTUR
XPA	R D	COORDINATES OF ELEMENT CORNER POINTS, X	SDATA
XPAD	R D	COORDINATES OF ELEMENT CORNER POINTS (DEFLECTED)	CONTRL
XPDE	R U	X-COORDINATE (DEFLECTED)	CONTRL
XR	R U	RIGHT-MOST LIMIT OF THE GRID ON X-AXIS	PLOT
XRG	R U	VALUE OF RIGHT SIDE OF HORIZONTAL SCALE	PICTUR
XS	R U	SURFACE X-COORDINATE POINT	ANALY2
XSC	R U	X SCALE FACTOR	PICTUR
XSC	R U	X SCALE FACTOR	SDATA
XSEP	R U	DISTANCE FROM LEADING EDGE MINUS UPSTREAM INTERACTION LENGTH	FLOSEP
XSEPP	R A	X-COORDINATE AT FLOW SEPARATION POINT	FLOSEP
XSEPP	R U	X-COORDINATE AT FLOW SEPARATION POINT	FORCE
XTE	R U	AVERAGE X-COORDINATE AT TRAILING EDGE	FLOSEP
XUO	R U	BOUNDARY CURVE POINT, X(U,O)	ANALY2

SYMBOL	TYPE	DEFINITION	ROUTINE
XU1	R U	BOUNDARY CURVE POINT, X(U,1)	ANALY2
XX	R U	X-COORDINATE	ANALY1
XX	R U	X-COORDINATE	ANALY2
XX	R U	X-COORDINATE	PICTUR
XX	R U	X-COORDINATE	SDATA
XX	R U	X-COORDINATE	SLABD
XXS	R U	X-COORDINATE	ANALY2
X0	R U	CENTER OF GRAVITY LOCATION	PLUNGE
X1	R A	X-COORDINATE	PUNCH
X1V00	R U	END POINT DERIVATIVE	ANALY2
X1V01	R U	END POINT DERIVATIVE	ANALY2
X1W	R U	BOUNDARY CURVE POINT, X(1,W)	ANALY2
X2	R A	X-COORDINATE	PUNCH
X2X1	R U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
X3X1	R U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
X3X2	R U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
Y	R U	Y-COORDINATE	ANALY1
Y	R U	Y-COORDINATE	ANALY2
Y	R U	PARAMETER IN CUBIC EQUATION	COMPR
Y	R U	Y-COORDINATE	PICTUR
Y	R D	PLOTTING ARRAY, LOCATION ALONG Y-AXIS	PLOT
Y	R U	Y-COORDINATE	SDATA
Y	R U	Y-COORDINATE	SLABD
YA	R U	Y-COORDINATE	ANALY1
YA	R D	Y-COORDINATE	ANALY2
YA	R D	Y-COORDINATE	PICTUR
YA	R D	Y-COORDINATE	SDATA
YA	R U	Y-COORDINATE	SLABD
YAVG	R U	AVERAGE Y COORDINATE	SDATA
YB	R U	Y-COORDINATE	ANALY1
YB	R D	Y-COORDINATE	ANALY2
YB	R D	Y-COORDINATE	PICTUR
YB	R U	BOTTOM MOST LIMIT OF THE GRID ON Y-AXIS	PLOT
YB	R D	Y-COORDINATE	SDATA
YBG	R U	VALUE OF BOTTOM OF VERTICAL SCALE	PICTUR
YB1	R D	BOUNDARY CURVE Y-COORDINATE ARRAY	ANALY2

SYMBOL	TYPE		DEFINITION	ROUTINE
YC	R	U	Y-COORDINATE	ANALY1
YCENT	R	U	QUADRILATERAL ELEMENT CENTROID-Y	CONTRL
YCENT	R	A	(NOT USED)	FLOSEP
YCENT	R	U	QUADRILATERAL ELEMENT CENTROID-Y	FORCE
YCENT	R	U	ELEMENT CENTROID COORDINATE-Y	PICTUR
YCENT	R	U	ELEMENT CENTROID COORDINATE-Y	SDATA
YCENT	R	U	ACTION POINT FOR FORCE VECTOR-Y	VECTOR
YCENTD	R	U	QUADRILATERAL ELEMENT CENTROID-Y (CONTROL DEFLECTED)	CONTRL
YCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-Y	AERO
YCENT2	R	C	ELEMENT CENTROID ARRAY-Y	COMPR
YCENT2	R	C	ELEMENT CENTROID ARRAY-Y	CONE
YCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Y	CONTRL
YCENT2	R	C	ELEMENT CENTROID COORDINATE-Y	DELWNG
YCENT2	R	C	ELEMENT CENTROID ARRAY-Y	EXPAND
YCENT2	R	C	(NOT USED)	FLOSEP
YCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Y	FORCE
YCENT2	R	C	ELEMENT CENTROID COORDINATE-Y	NEWTPM
YCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Y	QC
YCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-Y	SDATA
YCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Y	SHKEXP
YCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Y	SKINFR
YCENT2	R	C	QUADRILATERAL CENTROID ARRAY-Y	TEMP
YCG	R	U	Y-CENTER FOR MOMENT CALCULATIONS	AERO
YCG	R	A	Y-CENTER FOR MOMENT CALCULATIONS	FORCE
YCG	R	A	Y-CENTER FOR MOMENT CALCULATIONS	VECTOR
YD	R	U	Y-COORDINATE	ANALY1
YDIF	R	U	COORDINATE DIFFERENCE-Y	PICTUR
YDIF	R	U	COORDINATE DIFFERENCE-Y	SDATA
YHL	R	U	ELEMENT AVERAGE Y-COORDINATE AT HINGE LINE	FLOSEP
YHL1	R	U	HINGE LINE Y-COORDINATE OF POINT 1	CONTRL
YHL4	R	U	HINGE LINE Y-COORDINATE OF POINT 4	CONTRL
YIN	R	D	ELEMENT COORDINATES-Y	PICTUR
YIN	R	D	ELEMENT COORDINATE-Y	SDATA
YIN2	R	D	Y-COORDINATE FOR PLOT	PICTUR
YLE	R	U	Y-DISTANCE TO THE LEADING EDGE	SLABD
YLECL	R	U	Y-DISTANCE TO LEADING EDGE CENTER LINE	SLABD

SYMBOL	TYPE		DEFINITION	ROUTINE
YLEP1	R	U	SAVED Y-COORDINATE	SDATA
YOP	R	U	Y IN TRANSFORMED SYSTEM	CONTRL
YOPDE	R	U	Y IN TRANSFORMED SYSTEM (CONTROL DEFLECTED)	CONTRL
YOPH	R	U	Y-COORDINATE	CONTRL
YOW	R	U	BOUNDARY CURVE POINT, Y(O,W)	ANALY2
YO1	R	U	Y-COORDINATE FOR PLOT-POINT 1	PICTUR
YO2	R	U	Y-COORDINATE FOR PLOT-POINT 2	PICTUR
YO3	R	U	Y-COORDINATE FOR PLOT-POINT 3	PICTUR
YO4	R	U	Y-COORDINATE FOR PLOT-POINT 4	PICTUR
YP	R	U	Y-COORDINATE	CONTRL
YPA	R	D	COORDINATES OF ELEMENT CORNER POINTS, Y	CONTRL
YPA	R	A	Y-COORDINATES OF QUADRILATERAL ELEMENT	FLOSEP
YPA	R	D	Y-COORDINATES OF QUADRILATERAL ELEMENT	FORCE
YPA	R	U	COORDINATE OF ELEMENT CORNER POINT	PICTUR
YPA	R	D	COORDINATES OF ELEMENT CORNER POINTS-Y	SDATA
YPAD	R	D	COORDINATES OF ELEMENT CORNER POINTS (DEFLECTED)	CONTRL
YPDE	R	U	Y-COORDINATE (DEFLECTED)	CONTRL
YS	R	U	SURFACE Y-COORDINATE POINT	ANALY2
YSC	R	U	Y-SCALE FACTOR	PICTUR
YSC	R	U	Y-SCALE FACTOR	SDATA
YT	R	U	TOP MOST LIMIT OF THE GRID ON Y-AXIS	PLOT
YTE	R	U	AVERAGE Y-COORDINATE AT TRAILING EDGE	FLOSEP
YTG	R	U	VALUE OF TOP OF VERTICAL SCALE	PICTUR
YUO	R	U	BOUNDARY CURVE POINT, Y(U,O)	ANALY2
YU1	R	U	BOUNDARY CURVE POINT, Y(U,1)	ANALY2
YY	R	U	Y-COORDINATE	ANALY1
YY	R	U	Y-COORDINATE	ANALY2
YY	R	U	Y-COORDINATE	PICTUR
YY	R	U	Y-COORDINATE	SDATA
YY	R	U	Y-COORDINATE	SLABD
YYS	R	U	Y-COORDINATE	ANALY2
Y1	R	A	Y-COORDINATE	PUNCH
Y1VOO	R	U	END POINT DERIVATIVE	ANALY2
Y1VO1	R	U	END POINT DERIVATIVE	ANALY2
Y1W	R	U	BOUNDARY CURVE POINT, Y(1,W)	ANALY2
Y2	R	U	SECOND FUNCTION OF ODD EXPONENTIAL	BLUNT

SYMBOL	TYPE		DEFINITION	ROUTINE
Y2	R	A	Y-COORDINATE	PUNCH
Y2Y1	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
Y3Y1	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
Y3Y2	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
Z	R	U	Z-COORDINATE	ANALY1
Z	R	U	Z-COORDINATE	ANALY2
Z	R	U	GEOMETRIC ALTITUDE, FEET	ATMOS
Z	R	U	PARAMETER IN CUBIC EQUATION	COMPR
Z	R	U	FLOW CHARACTERISTIC PARAMETERS	EXPAND
Z	R	U	Z-COORDINATE	PICTUR
Z	R	U	Z-COORDINATE	SDATA
Z	R	U	Z-COORDINATE	SLABD
ZA	R	U	Z-COORDINATE	ANALY1
ZA	R	D	Z-COORDINATE	ANALY2
ZA	R	D	Z-COORDINATE	PICTUR
ZA	R	D	Z-COORDINATE	SDATA
ZA	R	U	Z-COORDINATE	SLABD
ZAVG	R	U	AVERAGE Z COORDINATE	SDATA
ZB	R	U	Z-COORDINATE	ANALY1
ZB	R	D	Z-COORDINATE	ANALY2
ZB	R	D	Z-COORDINATE	PICTUR
ZB	R	D	Z-COORDINATE	SDATA
ZB	R	D	Z-COORDINATE	SLABD
ZBI	R	D	BOUNDARY CURVE Z-COORDINATE ARRAY	ANALY2
ZC	R	U	Z-COORDINATE	ANALY1
ZCENT	R	U	QUADRILATERAL ELEMENT CENTROID-Z	CONTRL
ZCENT	R	A	(NOT USED)	FLOSEP
ZCENT	R	U	QUADRILATERAL ELEMENT CENTRAID-Z	FORCE
ZCENT	R	U	ELEMENT CENTROID COORDINATE-Z	PICTUR
ZCENT	R	U	ELEMENT CENTROID COORDINATE-Z	SDATA
ZCENT	R	U	ACTION POINT FOR FORCE VECTOR-Z	VECTOR
ZCENTD	R	U	QUADRILATERAL ELEMENT CENTROID-Z (CONTROL DEFLECTED)	CONTRL
ZCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-Z	AERO
ZCENT2	R	C	ELEMENT CENTROID ARRAY-Z	COMPR
ZCENT2	R	C	ELEMENT CENTROID ARRAY-Z	CONE
ZCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY, Z	CONTRL

SYMBOL	TYPE		DEFINITION	ROUTINE
ZCENT2	R	C	ELEMENT CENTROID COORDINATE-Z	DELWNG
ZCENT2	R	C	ELEMENT CENTROID ARRAY-Z	EXPAND
ZCENT2	R	C	(NOT USED)	FLOSEP
ZCENT2	R	C	QUADRILATERAL ELEMENT CENTRAID ARRAY-Z	FORCE
ZCENT2	R	C	ELEMENT CENTROID COORDINATE-Z	NEWTPM
ZCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Z	QC
ZCENT2	R	C	ELEMENT CENTROID COORDINATE ARRAY-Z	SDATA
ZCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Z	SHKEXP
ZCENT2	R	C	QUADRILATERAL ELEMENT CENTROID ARRAY-Z	SKINFR
ZCENT2	R	C	QUADRILATERAL CENTROID ARRAY-Z	TEMP
ZCG	R	U	Z-CENTER FOR MOMENT CALCULATIONS	AERO
ZCG	R	A	Z-CENTER FOR MOMENT CALCULATIONS	FORCE
ZCG	R	A	Z-CENTER FOR MOMENT CALCULATIONS	VECTOR
ZD	R	U	Z-COORDINATE	ANALY1
ZDIF	R	U	COORDINATE DIFFERENCE-Z	PICTUR
ZDIF	R	U	COORDINATE DIFFERENCE Z	SDATA
ZFACT	R	U	THICKNESS CORRECTION FACTOR FROM TABLES	SLABD
ZHL	R	U	ELEMENT AVERAGE Z-COORDINATE AT HINGE LINE	FLOSEP
ZHL1	R	U	HINGE LINE Z-COORDINATE OF POINT 1	CONTRL
ZHL4	R	U	HINGE LINE Z-COORDINATE OF POINT 4	CONTRL
ZIN	R	D	ELEMENT COORDINATES-Z	PICTUR
ZIN	R	D	ELEMENT COORDINATE-Z	SDATA
ZIN2	R	D	Z-COORDINATE FOR PLOT	PICTUR
ZLEP1	R	U	SAVED Z-COORDINATE	SDATA
ZLZ	R	U	INTERIM CALCULATION FOR PRESSURE EQUATION.	ATMOS
ZM	R	D	MATRIX OF GEOMETRIC ALTITUDES, FEET, ABOVE 245276 FEET	ATMOS
ZOP	R	U	Z IN TRANSFORMED SYSTEM	CONTRL
ZOPDE	R	U	Z IN TRANSFORMED SYSTEM (CONTROL DEFLECTED)	CONTRL
ZOPH	R	U	Z-COORDINATE	CONTRL
ZOW	R	U	BOUNDARY CURVE POINT, Z(O,W)	ANALY1
ZO1	R	U	Z-COORDINATE FOR PLOT-POINT 1	PICTUR
ZO2	R	U	Z-COORDINATE FOR PLOT-POINT 2	PICTUR
ZO3	R	U	Z-COORDINATE FOR PLOT-POINT 3	PICTUR
ZO4	R	U	Z-COORDINATE FOR PLOT-POINT 4	PICTUR
ZP	R	U	Z-COORDINATE	CONTRL
ZPA	R	D	COORDINATES OF ELEMENT CORNER POINTS, Z	CONTRL

SYMBOL	TYPE		DEFINITION	ROUTINE
ZPA	R	A	Z-COORDINATES OF QUADRILATERAL ELEMENT	FLOSEP
ZPA	R	D	Z-COORDINATES OF QUADRILATERAL ELEMENT	FORCE
ZPA	R	U	COORDINATE OF ELEMENT CORNER POINT	PICTUR
ZPA	R	D	COORDINATES OF ELEMENT CORNER POINTS-Z	SDATA
ZPAD	R	D	COORDINATES OF ELEMENT CORNER POINTS (DEFLECTED)	CONTRL
ZPDE	R	U	Z-COORDINATE (DEFLECTED)	CONTRL
ZS	R	U	SURFACE Z-COORDINATE POINT	ANALY2
ZSC	R	U	Z-SCALE FACTOR	PICTUR
ZSC	R	U	Z-SCALE FACTOR	SDATA
ZT	R	D	THICKNESS CORRECTION TABLE ARRAY	SLABD
ZTE	R	U	ELEMENT AVERAGE Z-COORDINATE AT TRAILING EDGE	FLOSEP
ZU0	R	U	BOUNDARY CURVE POINT, Z(U,0)	ANALY2
ZU1	R	U	BOUNDARY CURVE POINT, Z(U,1)	ANALY2
ZZ	R	U	Z-COORDINATE	ANALY1
ZZ	R	U	Z-COORDINATE	ANALY2
ZZ	R	U	Z-COORDINATE	PICTUR
ZZ	R	U	Z-COORDINATE	SDATA
ZZ	R	U	Z-COORDINATE	SLABD
ZZS	R	U	Z-COORDINATE	ANALY2
Z1	R	A	Z-COORDINATE	PUNCH
Z1V00	R	U	END POINT DERIVATIVE	ANALY2
Z1V01	R	U	END POINT DERIVATIVE	ANALY2
Z1W	R	U	BOUNDARY CURVE COORDINATE, Z(1,W)	ANALY2
Z2	R	A	Z-COORDINATE	PUNCH
Z221	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
Z321	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2
Z322	R	U	VARIABLE IN TANGENT VECTOR EQUATIONS	ANALY2

APPENDIX C

PROGRAM ARRAYS

This program contains a number of subscripted variable arrays. Most of these are single parameter arrays and are described in Appendix B. However, some of the arrays are used to represent several different program parameters. The most important of these are listed below. All of these arrays are used in the skin friction calculations.

Array Item	Description
ANGLE(1)	Angle through which flow is compressed or expanded. For an upper surface, + for expansion - for compression For a lower surface, + for compression - for expansion
ANGLE(2)	Compression or expansion angle (absolute value of ANGLE(1))
ANGLE(3)	Shock angle for compression and Mach angle for expansion
TR(1)	Altitude
TR(2)	Mach number
TR(3)	Velocity
TR(4)	Angle of attack of flight reference plane
TR(5)	Wall temperature, degrees Rankine, laminar
TR(6)	Wall temperature, degrees Rankine, turbulent
TR(7)	Wall enthalpy, laminar
TR(8)	Wall enthalpy, turbulent
TR(9)	Adiabatic wall enthalpy, laminar
TR(10)	Adiabatic wall enthalpy, turbulent
FS(I)	Flow conditions before shock or expansion
FS(1)	Density, slugs/ft^3
FS(2)	Pressure, pounds/ft^2
FS(3)	Temperature, degrees Rankine
FS(4)	Speed of sound, feet/sec
FS(5)	Viscosity, slugs/ft-sec

Array Item	Description
FS(6)	Mach number
FS(7)	Velocity, feet/sec
FS(8)	Reynolds number per foot
BS(I)	Flow conditions behind shock or expansion. See FS above for individual parameters.
SCF(1)	Total skin friction coefficient based on free stream properties and reference area. This is the sum of the proper combination of laminar and turbulent coefficients. Value in the lift direction.
SCF(2)	Total skin friction coefficient, value in the drag direction.
CFL(I)	Laminar skin friction values.
CFL(1)	Local average coefficient based on incompressible relations.
CFL(2)	Local average coefficient based on local flow conditions.
CFL(3)	Free stream skin friction coefficient based on local length.
CFL(4)	Total skin friction coefficient based on vehicle reference area.
CFL(5)	Total skin friction coefficient in the lift direction.
CFL(6)	Total skin friction coefficient in the drag direction.
CFT(I)	Turbulent skin friction values (see CFL above).
TS(1)	Reference temperature, degrees Rankine, laminar.
TS(2)	Reference temperature, degrees Rankine, turbulent.
RE(1)	Reference Reynolds number based on local length, laminar.
RE(2)	Reference Reynolds number based on local length, turbulent.
RF(1)	Recovery factor, laminar.
RF(2)	Recovery factor, turbulent.
RT(1)	Recovery temperature, laminar.
RT(2)	Recovery temperature, turbulent.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) McDonnell Douglas Corporation, Douglas Aircraft Division 3855 Lakewood Blvd. Long Beach, California 90801	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
HYPERSONIC ARBITRARY-BODY AERODYNAMIC COMPUTER PROGRAM MARK III VERSION
VOLUME II PROGRAM FORMULATION AND LISTINGS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final

5. AUTHOR(S) (Last name, first name, initial)
Gentry, Arvel E., and Smyth, Douglas N.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1968	661	65

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	DAC 61552, Vol II
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. AVAILABILITY/LIMITATION NOTICES
Distribution of this document is unlimited It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This report describes a digital computer program system that is capable of calculating the hypersonic aerodynamic characteristics of complex three-dimensional shapes. The outstanding features of this program are its flexibility in covering a very wide variety of problems and the multitude of program options available. The program is a combination of techniques and capabilities necessary in performing a complete aerodynamic analysis of hypersonic shapes. These include vehicle geometry generation and description, visual graphics necessary in handling geometry data and in preparing plots of the final aerodynamic data, aerodynamic calculations of surface pressures and skin friction forces, and the integration of these forces to give all aerodynamic coefficients and stability derivatives.

The geometric description techniques in this program provide the capability of handling completely arbitrary three-dimensional shapes. The procedure developed to check the accuracy of the geometric data uses a computer and automatic recorder to draw pictures of the vehicle viewed from any angle.

The pressure calculation methods provided within the program include modified Newtonian, blunt-body Newtonian-Prandtl-Meyer, tangent-wedge, tangent-cone, shock-expansion, Prandtl-Meyer expansion, blast wave, modified tangent-cone, boundary-layer induced pressures, free-molecular flow, and a number of empirical relationships. The pressure calculation method most suitable for each component of the vehicle is specified by the aerodynamicist. Viscous forces are also calculated and include viscous-inviscid interaction effects. Skin friction options include the Reference Temperature and the Reference Enthalpy methods (for both laminar and turbulent flow), the Spalding-Chi method (turbulent), and a special blunt body skin friction method. Control surface deflection pressures, including separation effects that may be caused by the deflected surface, are also calculated.

The program has been used to study a wide variety of hypersonic vehicle shapes including hypersonic cruise aircraft, air-breathing booster aircraft, blunt lifting reentry bodies, high L/D reentry vehicles, blunt reentry capsules, rocket boosters, reentry warheads, and satellite shapes.

The program is documented in two volumes. Volume I is primarily a User's Manual, and Volume II contains the Program Formulation and Listings.

DD FORM 1 JAN 64 1473

Unclassified
Security Classification

Unclassified

Security Classification

14 KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Aerodynamics Arbitrary Three-Dimensional Shapes Hypersonic Flow Computers (Digital) FORTRAN Graphics						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.